# 公共类

## 公共类

### 公共类 CLS\_AwardItem 奖励物品

[Desc("类型 对应 EAwardType")]

public short Type = 0;

[Desc("奖励配置ID")]

public int ConfigId = 0;

[Desc("奖励数量")]

public long Count = 0;

[Desc("等级")]

public int Level = 0;

### 公共类 CLS\_CountInfo 当前数量和上限数量

[Desc("当前")]

public long Value = 0;

[Desc("上限")]

public long Limit = 0;

### 公共类 CLS\_NeedInfo 需求数量和当前数量

[Desc("需求")]

public long Need = 0;

[Desc("当前")]

public long Value = 0;

### 公共类 CLS\_ItemNeedInfo 道具需求信息

[Desc("ID(配置ID)")]

public int Id = 0;

[Desc("数量")]

public int Count = 0;

[Desc("需求数量")]

public int CountNeed = 0;

### 公共类 CLS\_PubPlayerBase 玩家公共信息

[Desc("玩家唯一ID")]

public long Puid = 0;

[Desc("名字")]

public string Name = "";

[Desc("头像序号")]

public int HeadIndex = 0;

[Desc("所属州ID")]

public int StateId = 0;

[Desc("等级")]

public int Level = 0;

[Desc("所属势力唯一ID")]

public long GuildUid = 0;

[Desc("势力官职 EGuildOffice")]

public int GuildOffice = 0;

[Desc("战斗力")]

public int CombatPower = 0;

[Desc("上次登录时间")]

public long TickTimeLastLogin = 0;

### 公共类 CLS\_DataSnap 快照数据

[Desc("玩家唯一ID")]

public long Puid = 0;

[Desc("服务器ID")]

public int ServerID = 0;

[Desc("昵称")]

public string Name = "";

[Desc("头像序号(序号为0时使用图像数据)")]

public int HeadIndex = 0;

[Desc("所属州ID")]

public int StateId = 0;

[Desc("元宝")]

public long Gem = 0;

[Desc("等级")]

public int Level = 0;

[Desc("上次登录时间")]

public long TickTimeLastLogin = 0;

[Desc("活跃度状态 见EActivityLevel")]

public int ActivityLevel = 0;

[Desc("所属府ID")]

public long Mansion = 0;

[Desc("所属势力唯一ID")]

public long GuildUid = 0;

[Desc("势力官职 EGuildOffice")]

public int GuildOffice = 0;

[Desc("贡献")]

public int Contribution = 0;

[Desc("历史贡献")]

public int ContributionTotal = 0;

[Desc("战斗力")]

public int CombatPower = 0;

[Desc("主角加竞技场武将战力")]

public int AllArenaCombatPower = 0;

[Desc("玩家单条信息")]

public CLS\_PlayerData PlayerData = new CLS\_PlayerData();

### 公共类 CLS\_PlayerData 玩家单条信息

[Desc("玩家唯一ID")]

public long Puid = 0;

[Desc("头像序号(序号为0时使用图像数据)")]

public int HeadIndex = 0;

[Desc("名字")]

public string Name = "";

[Desc("分数")]

public long Score = 0;

[Desc("等级")]

public int Level = 0;

[Desc("防守队伍信息<位置，信息>")]

public Dictionary<int, CLS\_WarriorInfo> DictArenaWarrior = new Dictionary<int, CLS\_WarriorInfo>();

[Desc("个人签名")]

public string Signature = "";

[Desc("势力名称")]

public string GuildName = "";

### 公共类 CLS\_GameLog 动态日志

public DateTime GameLogTime = DateTime.Now;

[Desc("动态日志类型 EGameLogType")]

public int GameLogType = 0;

[Desc("参数")]

public List<string> Args = new List<string>();

# 通用协议 1-99

## 通用协议 All 10+

### 公共类 CLS\_BaseDemo 类示例

public byte a1 = 0;

public short a2 = 0;

public int a3 = 0;

public long a4 = 0;

public float a5 = 0.0f;

public string a6 = "";

public bool a7 = false;

public DateTime d1 = DateTime.Now;

public TimeSpan d2 = new TimeSpan();

public EnumBaseDemo BaseDemo = new EnumBaseDemo();

### 协议类 All\_Base\_Demo 示例

对应ID： ALL\_BASE\_DEMO

public byte a1 = 0;

public short a2 = 0;

public int a3 = 0;

public long a4 = 0;

public float a5 = 0.0f;

public string a6 = "";

public bool a7 = false;

public DateTime d1 = DateTime.Now;

public TimeSpan d2 = new TimeSpan();

public EnumBaseDemo eBaseDemo = new EnumBaseDemo();

public CLS\_BaseDemo BaseDemo = new CLS\_BaseDemo();

### 协议类 All\_Base\_Result 结果返回

对应ID： ALL\_BASE\_RESULT

[Desc("协议ID")]

public int HandleId = 0;

### 协议类 All\_Base\_Ping Ping服务器

对应ID： ALL\_BASE\_PING

[Desc("服务器时间刻度")]

public DateTime ServerTime = DateTime.Now;

### 协议类 All\_Base\_GameVersion 同步版本号

对应ID： ALL\_BASE\_GAMEVERSION

[Desc("同步版本号")]

public string GameVersion = "";

# 客户端调试

### 协议类 C2G\_Debug\_Cmd 请求 调试指令

对应ID： C2G\_DEBUG\_CMD

public int Cmd = 0;

public List<string> ListArgs = new List<string>();

### 协议类 G2C\_Debug\_Cmd 返回 调试指令

对应ID： G2C\_DEBUG\_CMD

## 角色登录和创建

### 协议类 G2C\_Login\_Connect 连接成功

对应ID： G2C\_LOGIN\_CONNECT

### 协议类 C2L\_Login\_UserLogin 请求 登陆用户账号

对应ID： C2L\_LOGIN\_USERLOGIN

[Desc("账号")]

public string Account = "";

[Desc("密码")]

public string Password = "";

### 协议类 L2C\_Login\_UserLogin 返回 登陆用户账号

对应ID： L2C\_LOGIN\_USERLOGIN

[Desc("用户id")]

public long Uuid = 0;

[Desc("验证码")]

public long Code = 0;

[Desc("是否需要显示协议")]

public bool NeedAgree = false;

### 协议类 C2G\_Login\_PlayerLogin 请求 登陆玩家

对应ID： C2G\_LOGIN\_PLAYERLOGIN

[Desc("用户id")]

public long Uuid = 0;

[Desc("验证码")]

public long Code = 0;

### 协议类 G2C\_Login\_PlayerLogin 返回 登陆玩家

对应ID： G2C\_LOGIN\_PLAYERLOGIN

### 协议类 C2G\_Login\_PlayerCreate 请求 玩家创建

对应ID： C2G\_LOGIN\_PLAYERCREATE

[Desc("玩家名字")]

public string Name = "";

[Desc("头像")]

public int Head = 0;

[Desc("所属州ID")]

public int StateId = 0;

### 协议类 G2C\_Login\_PlayerCreate 返回 玩家创建

对应ID： G2C\_LOGIN\_PLAYERCREATE

### 协议类 G2C\_Login\_Replaced 通知玩家被取代连接

对应ID： G2C\_LOGIN\_REPLACED

### 协议类 G2C\_Login\_Ban 通知玩家被封号

对应ID： G2C\_LOGIN\_BAN

### 公共类 CLS\_LoginStateInfo 州信息

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("人口数")]

public long Population = 0;

### 协议类 C2G\_Login\_ListState 请求 州信息

对应ID： C2G\_LOGIN\_LISTSTATE

### 协议类 G2C\_Login\_ListState 返回 州信息

对应ID： G2C\_LOGIN\_LISTSTATE

[Desc("州信息")]

public List<CLS\_LoginStateInfo> ListState = new List<CLS\_LoginStateInfo>();

### 协议类 C2G\_Login\_CheckName 请求 检测名字

对应ID： C2G\_LOGIN\_CHECKNAME

[Desc("玩家名字")]

public string Name = "";

### 协议类 G2C\_Login\_CheckName 返回 检测名字

对应ID： G2C\_LOGIN\_CHECKNAME

## 信息

### 公共类 CLS\_Info\_Base 玩家基本信息

[Desc("玩家唯一ID")]

public long Puid = 0;

[Desc("名字")]

public string Name = "";

[Desc("头像序号(序号为0时使用图像数据)")]

public int HeadIndex = 0;

[Desc("所属州ID")]

public int StateId = 0;

[Desc("元宝")]

public long Gem = 0;

[Desc("金币")]

public long Gold = 0;

[Desc("府玉")]

public long FuYu = 0;

[Desc("银两")]

public long Silver = 0;

[Desc("木材")]

public long Wood = 0;

[Desc("铁矿")]

public long Iron = 0;

[Desc("粮草")]

public long Food = 0;

[Desc("木材容量")]

public long CapacityWood = 0;

[Desc("铁矿容量")]

public long CapacityIron = 0;

[Desc("粮草容量")]

public long CapacityFood = 0;

[Desc("等级")]

public int Level = 0;

[Desc("经验值")]

public int Exp = 0;

[Desc("VIP等级")]

public int VipLevel = 0;

[Desc("Vip经验值")]

public int VipExp = 0;

[Desc("体力")]

public int Spirit = 0;

[Desc("挑战令牌")]

public int TowerToken = 0;

[Desc("武神积分")]

public int TowerScore = 0;

[Desc("军牌")]

public int Card = 0;

[Desc("声望")]

public int Prestige = 0;

[Desc("宴会厅积分")]

public int DrawScore = 0;

[Desc("所属势力唯一ID")]

public long GuildUid = 0;

[Desc("势力官职 EGuildOffice")]

public int GuildOffice = 0;

[Desc("所属势力名字")]

public string GuildName = "";

[Desc("势力都城")]

public int MyCapitalCity = 0;

[Desc("势力所属州")]

public int GuildState = 0;

[Desc("预备步兵")]

public int Soldiers1 = 0;

[Desc("预备骑兵")]

public int Soldiers2 = 0;

[Desc("预备弓兵")]

public int Soldiers3 = 0;

[Desc("购买金币次数")]

public int BuyGoldTimes = 0;

### 公共类 CLS\_Info\_Misc 玩家杂项信息

[Desc("客户端特殊标记")]

public Dictionary<int, int> dictFlagClient = new Dictionary<int, int>();

[Desc("木材已满")]

public bool WoodFull = false;

[Desc("铁矿已满")]

public bool IronFull = false;

[Desc("粮草已满")]

public bool FoodFull = false;

[Desc("可以签到")]

public bool CanSign = false;

[Desc("新手福利已完")]

public bool NoviceWelfareOver = false;

[Desc("未读邮件数目")]

public int MailUnread = 0;

[Desc("长安夺宝活动状态 1 开启 0未开启")]

public int BankStatus = 0;

[Desc("武将背包当前已买ID")]

public int WarriorBagId = 0;

[Desc("当前武将数目")]

public int WarriorCount = 0;

[Desc("武将背包上限")]

public int WarriorCountMax = 0;

[Desc("新手福利数目")]

public int WelfareAwardNum = 0;

[Desc("个人签名")]

public string Signature = "";

[Desc("好友提示")]

public bool bFriend = false;

[Desc("私聊提示")]

public bool bChat = false;

[Desc("注册时间")]

public long TickTimeRegister = 0;

### 协议类 C2G\_Info\_GetAll 请求 玩家全部信息

对应ID： C2G\_INFO\_GETALL

### 协议类 G2C\_Info\_GetAll 返回 玩家全部信息

对应ID： G2C\_INFO\_GETALL

public CLS\_Info\_Base InfoBase = new CLS\_Info\_Base();

public CLS\_Info\_Misc InfoMisc = new CLS\_Info\_Misc();

[Desc("道具列表")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

### 协议类 G2C\_Info\_PushAll 推送 玩家信息 全部

对应ID： G2C\_INFO\_PUSHALL

public CLS\_Info\_Base InfoBase = new CLS\_Info\_Base();

public CLS\_Info\_Misc InfoMisc = new CLS\_Info\_Misc();

### 协议类 G2C\_Info\_PushItem 推送 玩家信息 道具

对应ID： G2C\_INFO\_PUSHITEM

[Desc("道具列表")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

### 协议类 G2C\_Info\_PushEquip 推送 玩家信息 装备

对应ID： G2C\_INFO\_PUSHEQUIP

[Desc("页码1-n (返回每页100条) (0=结束)")]

public int PageIndex = 0;

[Desc("装备列表")]

public Dictionary<long, CLS\_EquipInfo> DictEquip = new Dictionary<long, CLS\_EquipInfo>();

### 协议类 G2C\_Info\_PushWarrior 推送 玩家信息 武将

对应ID： G2C\_INFO\_PUSHWARRIOR

[Desc("页码1-n (返回每页100条) (0=结束)")]

public int PageIndex = 0;

[Desc("武将列表")]

public List<CLS\_WarriorInfo> ListWarrior = new List<CLS\_WarriorInfo>();

### 公共类 CLS\_UpdateInfoItem 更新 道具

public byte Mode = 0;

public CLS\_ItemInfo ItemInfo = new CLS\_ItemInfo();

### 公共类 CLS\_UpdateInfoEquip 更新 装备

public byte Mode = 0;

public CLS\_EquipInfo EquipInfo = new CLS\_EquipInfo();

### 公共类 CLS\_UpdateInfoWarrior 更新 武将

public byte Mode = 0;

public CLS\_WarriorInfo WarriorInfo = new CLS\_WarriorInfo();

### 协议类 G2C\_Info\_PushUpdate 推送 玩家信息 更新

对应ID： G2C\_INFO\_PUSHUPDATE

public List<CLS\_UpdateInfoItem> ListUpdateItem = new List<CLS\_UpdateInfoItem>();

public List<CLS\_UpdateInfoEquip> ListUpdateEquip = new List<CLS\_UpdateInfoEquip>();

public List<CLS\_UpdateInfoWarrior> ListUpdateWarrior = new List<CLS\_UpdateInfoWarrior>();

### 协议类 C2G\_Info\_ChangeName 请求 更改姓名

对应ID： C2G\_INFO\_CHANGENAME

public string Name = "";

### 协议类 G2C\_Info\_ChangeName 返回 更改姓名

对应ID： G2C\_INFO\_CHANGENAME

public string Name = "";

### 协议类 C2G\_Info\_ChangeFlagClient 请求 更改客户端特殊标记

对应ID： C2G\_INFO\_CHANGEFLAGCLIENT

public int FlagKey = 0;

public int FlagValue = 0;

### 协议类 G2C\_Info\_ChangeFlagClient 返回 更改客户端特殊标记

对应ID： G2C\_INFO\_CHANGEFLAGCLIENT

public int FlagKey = 0;

public int FlagValue = 0;

### 协议类 C2G\_Info\_ChangeHeadIndex 请求 更换头像

对应ID： C2G\_INFO\_CHANGEHEADINDEX

[Desc("头像序号(序号为0时使用图像数据)")]

public int HeadIndex = 0;

### 协议类 G2C\_Info\_ChangeHeadIndex 返回 更换头像

对应ID： G2C\_INFO\_CHANGEHEADINDEX

[Desc("头像序号(序号为0时使用图像数据)")]

public int HeadIndex = 0;

### 协议类 C2G\_Info\_ChangeHeadData 请求 上传头像数据

对应ID： C2G\_INFO\_CHANGEHEADDATA

[Desc("头像数据")]

public List<byte> HeadData = new List<byte>();

### 协议类 G2C\_Info\_ChangeHeadData 返回 上传头像数据

对应ID： G2C\_INFO\_CHANGEHEADDATA

### 协议类 C2G\_Info\_SubmitBug 请求 反馈问题

对应ID： C2G\_INFO\_SUBMITBUG

public string Text = "";

### 协议类 G2C\_Info\_SubmitBug 返回 反馈问题

对应ID： G2C\_INFO\_SUBMITBUG

### 协议类 G2C\_Info\_NotifyLevelUp 通知 玩家升级

对应ID： G2C\_INFO\_NOTIFYLEVELUP

[Desc("现等级")]

public int Level = 0;

[Desc("现经验")]

public int Exp = 0;

[Desc("原等级")]

public int LevelOld = 0;

### 协议类 C2G\_Info\_PlayerData 请求 玩家单条信息

对应ID： C2G\_INFO\_PLAYERDATA

[Desc("唯一ID")]

public long Id = 0;

### 协议类 G2C\_Info\_PlayerData 返回 玩家单条信息

对应ID： G2C\_INFO\_PLAYERDATA

[Desc("玩家单条信息")]

public CLS\_PlayerData PlayerInfo = new CLS\_PlayerData();

[Desc("是否是好友")]

public bool bFriend = false;

[Desc("是否已屏蔽")]

public bool bScreen = false;

### 协议类 C2G\_Info\_ChangeSignature 请求 更改个人签名

对应ID： C2G\_INFO\_CHANGESIGNATURE

public string Signature = "";

### 协议类 G2C\_Info\_ChangeSignature 返回 更改个人签名

对应ID： G2C\_INFO\_CHANGESIGNATURE

public string Signature = "";

## 玩家进入游戏时弹出的游戏公告

### 协议类 C2G\_Notice\_System 请求 系统公告

对应ID： C2G\_NOTICE\_SYSTEM

### 协议类 G2C\_Notice\_System 返回 系统公告

对应ID： G2C\_NOTICE\_SYSTEM

[Desc("内容")]

public string Text = "";

### 协议类 C2G\_Notice\_Activity 请求 活动公告

对应ID： C2G\_NOTICE\_ACTIVITY

### 协议类 G2C\_Notice\_Activity 返回 活动公告

对应ID： G2C\_NOTICE\_ACTIVITY

[Desc("内容")]

public List<string> ListText = new List<string>();

### 协议类 G2C\_Notice\_Rolling 发送 走马灯

对应ID： G2C\_NOTICE\_ROLLING

[Desc("走马灯类型 见ENoticeRollingType")]

public byte Type = 0;

[Desc("内容")]

public string Text = "";

[Desc("播放次数")]

public int Count = 0;

## 邮件

### 公共类 CLS\_MailInfo 邮件信息

[Desc("邮件ID")]

public long Id = 0;

[Desc("邮件类型")]

public short MailType = 0;

[Desc("标题")]

public string Title = "";

[Desc("是否有附件")]

public bool HasAttachments = false;

[Desc("已读")]

public bool Readed = false;

[Desc("已领")]

public bool Got = false;

[Desc("保存")]

public bool Saved = false;

[Desc("发送时间")]

public string CreateTime = "";

[Desc("已发邮件")]

public bool Sended = false;

[Desc("收件人(已发邮件生效 个人邮件:名字 郡邮件:郡ID 府邮件:府ID)")]

public string ReceiveName = "";

[Desc("发件人名字")]

public string SenderName = "";

### 公共类 CLS\_MailInfoDetails 邮件信息 详细

[Desc("邮件ID")]

public long Id = 0;

[Desc("邮件信息")]

public CLS\_MailInfo MailInfo = new CLS\_MailInfo();

[Desc("正文")]

public string Body = "";

[Desc("附件列表")]

public List<CLS\_AwardItem> listAward = new List<CLS\_AwardItem>();

### 公共类 CLS\_MailInfoSend 邮件信息 发送

public short MailType = 0;

[Desc("标题")]

public string Title = "";

[Desc("正文")]

public string Body = "";

[Desc("发件人名字")]

public string SenderName = "";

[Desc("收件人名字")]

public string ReceiveName = "";

### 公共类 CLS\_SystemMailInfoSend 系统给玩家发送邮件信息

[Desc("邮件ID")]

public long Id = 0;

public short TargetType = 0;

[Desc("接收者列表")]

public List<string> ListTarget = new List<string>();

[Desc("标题")]

public string Title = "";

[Desc("正文")]

public string Body = "";

[Desc("邮件发送者")]

public string SenderName = "";

[Desc("附件列表")]

public List<CLS\_AwardItem> ListAttachments = new List<CLS\_AwardItem>();

### 协议类 C2G\_Mail\_List 请求 邮件列表

对应ID： C2G\_MAIL\_LIST

### 协议类 G2C\_Mail\_List 返回 邮件列表

对应ID： G2C\_MAIL\_LIST

[Desc("邮件列表")]

public List<CLS\_MailInfo> ListMail = new List<CLS\_MailInfo>();

### 协议类 C2G\_Mail\_Read 请求 邮件读取

对应ID： C2G\_MAIL\_READ

[Desc("邮件ID")]

public long Id = 0;

### 协议类 G2C\_Mail\_Read 返回 邮件读取

对应ID： G2C\_MAIL\_READ

[Desc("邮件ID")]

public long Id = 0;

[Desc("邮件信息详细")]

public CLS\_MailInfoDetails Details = new CLS\_MailInfoDetails();

### 协议类 C2G\_Mail\_Get 请求 邮件领取附件

对应ID： C2G\_MAIL\_GET

[Desc("邮件ID")]

public long Id = 0;

### 协议类 G2C\_Mail\_Get 返回 邮件领取附件

对应ID： G2C\_MAIL\_GET

[Desc("邮件ID")]

public long Id = 0;

[Desc("附件列表")]

public List<CLS\_AwardItem> listAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Mail\_GetAll 请求 邮件一键领取

对应ID： C2G\_MAIL\_GETALL

### 协议类 G2C\_Mail\_GetAll 返回 邮件一键领取

对应ID： G2C\_MAIL\_GETALL

[Desc("附件列表")]

public List<CLS\_AwardItem> listAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Mail\_Delete 请求 邮件删除

对应ID： C2G\_MAIL\_DELETE

[Desc("邮件ID")]

public long Id = 0;

### 协议类 G2C\_Mail\_Delete 返回 邮件删除

对应ID： G2C\_MAIL\_DELETE

[Desc("邮件ID")]

public long Id = 0;

### 协议类 C2G\_Mail\_DeleteReaded 请求 邮件清理已读

对应ID： C2G\_MAIL\_DELETEREADED

[Desc("要删除的当前页面类型")]

public short MailType = 0;

### 协议类 G2C\_Mail\_DeleteReaded 返回 邮件清理已读

对应ID： G2C\_MAIL\_DELETEREADED

[Desc("刷新后邮件列表")]

public List<CLS\_MailInfo> ListMail = new List<CLS\_MailInfo>();

### 协议类 C2G\_Mail\_Save 请求 邮件保存/取消保存

对应ID： C2G\_MAIL\_SAVE

[Desc("邮件ID")]

public long Id = 0;

### 协议类 G2C\_Mail\_Save 返回 邮件保存/取消保存

对应ID： G2C\_MAIL\_SAVE

[Desc("邮件ID")]

public long Id = 0;

### 协议类 C2G\_Mail\_Send 请求 邮件发送

对应ID： C2G\_MAIL\_SEND

[Desc("邮件发送信息")]

public CLS\_MailInfoSend MailInfoSend = new CLS\_MailInfoSend();

### 协议类 G2C\_Mail\_Send 返回 邮件发送

对应ID： G2C\_MAIL\_SEND

[Desc("新加的已发邮件列表")]

public CLS\_MailInfo MailInfo = new CLS\_MailInfo();

## 聊天

### 公共类 CLS\_ChatPlayerBase 聊天玩家信息

[Desc("玩家唯一ID")]

public long Puid = 0;

[Desc("名字")]

public string Name = "";

[Desc("头像序号")]

public int HeadIndex = 0;

### 公共类 CLS\_ChatMsg 聊天信息

[Desc("消息时间")]

public DateTime ChatTime = DateTime.Now;

[Desc("聊天频道 见EChatChannel")]

public short Channel = 0;

[Desc("发送者信息")]

public CLS\_ChatPlayerBase SenderInfo = new CLS\_ChatPlayerBase();

[Desc("文本")]

public string Text = "";

### 公共类 CLS\_ChatList 私聊记录

[Desc("对方信息")]

public CLS\_ChatPlayerBase Info = new CLS\_ChatPlayerBase();

public List<CLS\_ChatMsg> ListChatMsgs = new List<CLS\_ChatMsg>();

### 协议类 C2G\_Chat\_Send 请求 聊天信息发送

对应ID： C2G\_CHAT\_SEND

[Desc("聊天频道 见EChatChannel")]

public short Channel = 0;

[Desc("文本")]

public string Text = "";

[Desc("对方唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Chat\_Send 返回 聊天信息发送

对应ID： G2C\_CHAT\_SEND

### 协议类 G2C\_Chat\_Receive 通知 聊天信息接收

对应ID： G2C\_CHAT\_RECEIVE

[Desc("聊天信息列表")]

public List<CLS\_ChatMsg> ListChatMsgs = new List<CLS\_ChatMsg>();

### 协议类 C2G\_Chat\_GetPrivateChat 请求 请求私聊数据

对应ID： C2G\_CHAT\_GETPRIVATECHAT

### 协议类 G2C\_Chat\_GetPrivateChat 返回 聊天信息发送

对应ID： G2C\_CHAT\_GETPRIVATECHAT

public Dictionary<long, CLS\_ChatList> DictChatInfo = new Dictionary<long, CLS\_ChatList>();

### 协议类 C2G\_Chat\_Remove 请求 删除最近联系人

对应ID： C2G\_CHAT\_REMOVE

public long Uid = 0;

### 协议类 G2C\_Chat\_Remove 返回 删除最近联系人

对应ID： G2C\_CHAT\_REMOVE

public long Uid = 0;

## 好友

### 公共类 CLS\_FriendInfo 玩家单条信息

[Desc("玩家唯一ID")]

public long Uid = 0;

[Desc("头像")]

public int HeadIndex = 0;

[Desc("名字")]

public string Name = "";

[Desc("等级")]

public int Level = 0;

[Desc("个人签名")]

public string Signature = "";

[Desc("体力状态：1-可领取 2-可赠送 3-已赠送")]

public int GiveState = 0;

[Desc("是否在线")]

public bool bOnline = false;

### 协议类 C2G\_Friend\_List 请求 好友列表

对应ID： C2G\_FRIEND\_LIST

### 协议类 G2C\_Friend\_List 返回 好友列表

对应ID： G2C\_FRIEND\_LIST

[Desc("好友列表")]

public List<CLS\_FriendInfo> ListItem = new List<CLS\_FriendInfo>();

### 协议类 C2G\_Friend\_ListApply 请求 申请列表

对应ID： C2G\_FRIEND\_LISTAPPLY

### 协议类 G2C\_Friend\_ListApply 返回 申请列表

对应ID： G2C\_FRIEND\_LISTAPPLY

[Desc("申请列表")]

public List<CLS\_FriendInfo> ListItem = new List<CLS\_FriendInfo>();

### 协议类 C2G\_Friend\_ListRecommend 请求 推荐列表

对应ID： C2G\_FRIEND\_LISTRECOMMEND

### 协议类 G2C\_Friend\_ListRecommend 返回 推荐列表

对应ID： G2C\_FRIEND\_LISTRECOMMEND

[Desc("推荐列表")]

public List<CLS\_FriendInfo> ListItem = new List<CLS\_FriendInfo>();

### 协议类 C2G\_Friend\_Apply 请求 申请好友

对应ID： C2G\_FRIEND\_APPLY

[Desc("对象玩家唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Friend\_Apply 返回 申请好友

对应ID： G2C\_FRIEND\_APPLY

[Desc("对象玩家唯一ID")]

public long Uid = 0;

public bool bFriend = false;

### 协议类 C2G\_Friend\_Add 请求 添加好友

对应ID： C2G\_FRIEND\_ADD

[Desc("是否同意")]

public bool bAgree = false;

[Desc("是否全部")]

public bool bAll = false;

[Desc("对象玩家唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Friend\_Add 返回 添加好友

对应ID： G2C\_FRIEND\_ADD

### 协议类 C2G\_Friend\_Remove 请求 请求删除

对应ID： C2G\_FRIEND\_REMOVE

[Desc("对象玩家唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Friend\_Remove 返回 请求删除

对应ID： G2C\_FRIEND\_REMOVE

### 协议类 C2G\_Friend\_Give 请求 赠送体力

对应ID： C2G\_FRIEND\_GIVE

[Desc("全部赠送")]

public bool bAll = false;

[Desc("对象玩家唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Friend\_Give 返回 赠送体力

对应ID： G2C\_FRIEND\_GIVE

### 协议类 C2G\_Friend\_Receive 请求 领取体力

对应ID： C2G\_FRIEND\_RECEIVE

[Desc("全部领取")]

public bool bAll = false;

[Desc("对象玩家唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Friend\_Receive 返回 领取体力

对应ID： G2C\_FRIEND\_RECEIVE

### 协议类 C2G\_Friend\_Screen 请求 屏蔽好友

对应ID： C2G\_FRIEND\_SCREEN

[Desc("对象玩家唯一ID")]

public long Uid = 0;

[Desc("是否屏蔽")]

public bool bScreen = false;

### 协议类 G2C\_Friend\_Screen 返回 屏蔽好友

对应ID： G2C\_FRIEND\_SCREEN

[Desc("对象玩家唯一ID")]

public long Uid = 0;

[Desc("是否屏蔽")]

public bool bScreen = false;

## 世界地图

### 公共类 CLS\_MapCitySimple 城池信息简单

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("城池归属 EOwnership")]

public byte Ownership = 0;

### 协议类 C2G\_Map\_ListCity 请求 城池列表

对应ID： C2G\_MAP\_LISTCITY

### 协议类 G2C\_Map\_ListCity 返回 城池列表

对应ID： G2C\_MAP\_LISTCITY

[Desc("城池列表")]

public List<CLS\_MapCitySimple> ListCity = new List<CLS\_MapCitySimple>();

### 公共类 CLS\_MapCityInfo 城池信息

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("占据势力名")]

public string GuildName = "";

[Desc("太守名")]

public string LeaderName = "";

[Desc("繁荣度")]

public long Prosperity = 0;

### 协议类 C2G\_Map\_CityInfo 请求 城池信息

对应ID： C2G\_MAP\_CITYINFO

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

### 协议类 G2C\_Map\_CityInfo 返回 城池信息

对应ID： G2C\_MAP\_CITYINFO

[Desc("城池信息")]

public CLS\_MapCityInfo CityInfo = new CLS\_MapCityInfo();

### 公共类 CLS\_MapMyCityInfo 我方城池信息

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("繁荣度")]

public long Prosperity = 0;

### 协议类 C2G\_Map\_ListMyCity 请求 我方城池列表

对应ID： C2G\_MAP\_LISTMYCITY

### 协议类 G2C\_Map\_ListMyCity 返回 我方城池列表

对应ID： G2C\_MAP\_LISTMYCITY

[Desc("我方城池列表")]

public List<CLS\_MapMyCityInfo> ListMyCity = new List<CLS\_MapMyCityInfo>();

## 道具

### 公共类 CLS\_ItemInfo 道具信息

[Desc("ID(配置ID)")]

public int Id = 0;

[Desc("数量")]

public int Count = 0;

### 协议类 C2G\_Item\_List 请求 道具列表

对应ID： C2G\_ITEM\_LIST

### 协议类 G2C\_Item\_List 返回 道具列表

对应ID： G2C\_ITEM\_LIST

[Desc("道具列表")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

### 协议类 C2G\_Item\_Sell 请求 道具卖店

对应ID： C2G\_ITEM\_SELL

[Desc("ID(配置ID)")]

public int Id = 0;

[Desc("数量")]

public int Count = 0;

### 协议类 G2C\_Item\_Sell 返回 道具卖店

对应ID： G2C\_ITEM\_SELL

[Desc("道具列表")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

### 协议类 C2G\_Item\_Use 请求 道具使用

对应ID： C2G\_ITEM\_USE

[Desc("ID(配置ID)")]

public int Id = 0;

[Desc("类型0:使用，1:合成")]

public int Type = 0;

[Desc("数量")]

public int Count = 0;

### 协议类 G2C\_Item\_Use 返回 道具使用

对应ID： G2C\_ITEM\_USE

[Desc("类型0:使用，1:合成")]

public int Type = 0;

[Desc("道具列表")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

[Desc("奖励项")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Item\_ResourcesList 请求 资源道具列表

对应ID： C2G\_ITEM\_RESOURCESLIST

[Desc("道具类型 EUseEffect")]

public int Type = 0;

### 协议类 G2C\_Item\_ResourcesList 返回 资源道具列表

对应ID： G2C\_ITEM\_RESOURCESLIST

[Desc("道具列表")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

### 协议类 C2G\_Item\_ResourcesUse 请求 资源道具使用

对应ID： C2G\_ITEM\_RESOURCESUSE

[Desc("道具")]

public List<CLS\_ItemInfo> Items = new List<CLS\_ItemInfo>();

[Desc("道具类型 EUseEffect")]

public int Type = 0;

### 协议类 G2C\_Item\_ResourcesUse 返回 资源道具使用

对应ID： G2C\_ITEM\_RESOURCESUSE

[Desc("道具列表")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

[Desc("奖励项")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Item\_ResourcesBuy 请求 资源道具购买

对应ID： C2G\_ITEM\_RESOURCESBUY

[Desc("道具ID ShopSpeedUp表里的ID")]

public int ID = 0;

[Desc("数量")]

public int Amount = 0;

[Desc("道具类型 EUseEffect")]

public int Type = 0;

### 协议类 G2C\_Item\_ResourcesBuy 返回 资源道具购买

对应ID： G2C\_ITEM\_RESOURCESBUY

[Desc("道具列表")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

[Desc("奖励项")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Item\_CombatUse 请求 战斗使用道具

对应ID： C2G\_ITEM\_COMBATUSE

[Desc("ID(配置ID)")]

public int Id = 0;

[Desc("数量")]

public int Count = 0;

[Desc("转发战斗效果")]

public int CombatType = 0;

### 协议类 G2C\_Item\_CombatUse 返回 战斗使用道具

对应ID： G2C\_ITEM\_COMBATUSE

[Desc("ID(配置ID)")]

public int Id = 0;

[Desc("数量")]

public int Count = 0;

[Desc("转发战斗效果")]

public int CombatType = 0;

[Desc("道具列表")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

## 装备

### 公共类 CLS\_EquipSuffix 装备高级属性

[Desc("配置ID")]

public int ConfigId = 0;

[Desc("属性类型 见EPassiveType")]

public int PassiveType = 0;

[Desc("属性制")]

public float SuffixValue = 0.0f;

### 公共类 CLS\_EquipInfo 装备信息

[Desc("装备唯一ID")]

public long Id = 0;

[Desc("配置ID")]

public int ConfigId = 0;

[Desc("装备类型 见EEquipType")]

public int EquipType = 0;

[Desc("强化等级")]

public int Intensify = 0;

[Desc("战斗属性")]

public Dictionary<int, float> CombatProperty = new Dictionary<int, float>();

[Desc("自定义名称")]

public string MarkName = "";

public List<CLS\_EquipSuffix> SuffixProperty = new List<CLS\_EquipSuffix>();

### 协议类 C2G\_Equip\_List 请求 装备列表

对应ID： C2G\_EQUIP\_LIST

### 协议类 G2C\_Equip\_List 返回 装备列表

对应ID： G2C\_EQUIP\_LIST

[Desc("页码1-n (返回每页100条) (0=结束)")]

public int PageIndex = 0;

[Desc("装备列表")]

public Dictionary<long, CLS\_EquipInfo> DictEquip = new Dictionary<long, CLS\_EquipInfo>();

### 协议类 C2G\_Equip\_WarriorList 请求 武将装备列表

对应ID： C2G\_EQUIP\_WARRIORLIST

[Desc("武将唯一ID")]

public long Id = 0;

### 协议类 G2C\_Equip\_WarriorList 返回 武将装备列表

对应ID： G2C\_EQUIP\_WARRIORLIST

[Desc("武将已装备列表<装备位置, 信息>")]

public Dictionary<int, CLS\_EquipInfo> DictEquiped = new Dictionary<int, CLS\_EquipInfo>();

[Desc("武将装备列表")]

public Dictionary<long, CLS\_EquipInfo> DictEquip = new Dictionary<long, CLS\_EquipInfo>();

[Desc("武将现信息")]

public CLS\_WarriorInfo WarriorInfo = new CLS\_WarriorInfo();

### 协议类 C2G\_Equip\_WarriorAll 请求 所有武将装备列表

对应ID： C2G\_EQUIP\_WARRIORALL

### 协议类 G2C\_Equip\_WarriorAll 返回 所有武将装备列表

对应ID： G2C\_EQUIP\_WARRIORALL

[Desc("武将装备列表")]

public Dictionary<long, CLS\_EquipInfo> DictEquip = new Dictionary<long, CLS\_EquipInfo>();

### 协议类 C2G\_Equip\_WarriorWield 请求 武将装备运用

对应ID： C2G\_EQUIP\_WARRIORWIELD

[Desc("武将唯一ID")]

public long WarriorId = 0;

[Desc("装备唯一ID")]

public long Id = 0;

### 协议类 G2C\_Equip\_WarriorWield 返回 武将装备运用

对应ID： G2C\_EQUIP\_WARRIORWIELD

[Desc("武将已装备列表<装备位置, 信息>")]

public Dictionary<int, CLS\_EquipInfo> DictEquiped = new Dictionary<int, CLS\_EquipInfo>();

[Desc("武将装备列表")]

public Dictionary<long, CLS\_EquipInfo> DictEquip = new Dictionary<long, CLS\_EquipInfo>();

[Desc("武将现信息")]

public CLS\_WarriorInfo WarriorInfo = new CLS\_WarriorInfo();

### 协议类 C2G\_Equip\_WarriorRemove 请求 武将装备卸下

对应ID： C2G\_EQUIP\_WARRIORREMOVE

[Desc("武将唯一ID")]

public long WarriorId = 0;

[Desc("装备位置(装备类型)")]

public int EquipType = 0;

### 协议类 G2C\_Equip\_WarriorRemove 返回 武将装备卸下

对应ID： G2C\_EQUIP\_WARRIORREMOVE

[Desc("武将已装备列表<装备位置, 信息>")]

public Dictionary<int, CLS\_EquipInfo> DictEquiped = new Dictionary<int, CLS\_EquipInfo>();

[Desc("武将装备列表")]

public Dictionary<long, CLS\_EquipInfo> DictEquip = new Dictionary<long, CLS\_EquipInfo>();

[Desc("武将现信息")]

public CLS\_WarriorInfo WarriorInfo = new CLS\_WarriorInfo();

### 协议类 C2G\_Equip\_WarriorResolve 请求 武将装备分解

对应ID： C2G\_EQUIP\_WARRIORRESOLVE

[Desc("分解装备列表唯一ID")]

public List<long> Id = new List<long>();

### 协议类 G2C\_Equip\_WarriorResolve 返回 武将装备分解

对应ID： G2C\_EQUIP\_WARRIORRESOLVE

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Equip\_Rename 请求 装备修改名称

对应ID： C2G\_EQUIP\_RENAME

[Desc("装备自定义名称")]

public string MarkName = "";

### 协议类 G2C\_Equip\_Rename 返回 装备修改名称

对应ID： G2C\_EQUIP\_RENAME

### 协议类 C2G\_Equip\_IntensifyInfo 请求 装备强化信息

对应ID： C2G\_EQUIP\_INTENSIFYINFO

[Desc("装备ID")]

public long EquipID = 0;

### 协议类 G2C\_Equip\_IntensifyInfo 返回 装备强化信息

对应ID： G2C\_EQUIP\_INTENSIFYINFO

[Desc("强化信息")]

public CLS\_EquipIntensifyInfo IntensityInfo = new CLS\_EquipIntensifyInfo();

### 协议类 C2G\_Equip\_Intensify 请求 装备强化

对应ID： C2G\_EQUIP\_INTENSIFY

[Desc("装备ID")]

public long EquipID = 0;

### 协议类 G2C\_Equip\_Intensify 返回 装备强化

对应ID： G2C\_EQUIP\_INTENSIFY

[Desc("强化信息")]

public CLS\_EquipIntensifyInfo IntensityInfo = new CLS\_EquipIntensifyInfo();

[Desc("结果 对应EIntensifyResult")]

public int IntensifyResult = 0;

[Desc("产物")]

public List<CLS\_AwardItem> AwardItem = new List<CLS\_AwardItem>();

### 公共类 CLS\_EquipIntensifyInfo 装备强化信息

[Desc("强化等级")]

public int IntensityId = 0;

[Desc("攻击")]

public float Atk = 0.0f;

[Desc("防御")]

public float Def = 0.0f;

[Desc("兵力")]

public int Hp = 0;

[Desc("谋略")]

public float Inte = 0.0f;

[Desc("强化后攻击")]

public float IntensityAtk = 0.0f;

[Desc("强化后防御")]

public float IntensityDef = 0.0f;

[Desc("强化后谋略")]

public float IntensityInte = 0.0f;

[Desc("强化后兵力")]

public int IntensityHp = 0;

[Desc("银两消耗")]

public int Gold = 0;

[Desc("需求道具列表")]

public List<CLS\_ItemNeedInfo> ListItemNeedInfo = new List<CLS\_ItemNeedInfo>();

## 武将

### 公共类 CLS\_SkillInfo 技能信息

[Desc("技能ID(配置ID SkillId)")]

public int SkillId = 0;

[Desc("等级")]

public int Level = 0;

[Desc("经验值")]

public int Exp = 0;

### 公共类 CLS\_WarriorInfo 武将信息

[Desc("唯一ID")]

public long Id = 0;

[Desc("配置ID")]

public int ConfigId = 0;

[Desc("等级")]

public int Level = 0;

[Desc("经验")]

public int Exp = 0;

[Desc("武将自定义名称")]

public string MarkName = "";

[Desc("锁定")]

public bool Lock = false;

[Desc("募兵")]

public bool InRecruit = false;

[Desc("上阵")]

public bool InBattle = false;

[Desc("被俘虏")]

public bool Captive = false;

[Desc("被雇佣")]

public bool Employed = false;

[Desc("出征PVP")]

public bool InArmy = false;

[Desc("军务中")]

public bool InAffairs = false;

[Desc("矿点占领")]

public bool InGrab = false;

[Desc("武将技能列表")]

public List<CLS\_SkillInfo> ListSkill = new List<CLS\_SkillInfo>();

[Desc("进阶等级")]

public int AdvanceLv = 0;

[Desc("固定攻击")]

public float BaseAtk = 0.0f;

[Desc("固定防御")]

public float BaseDef = 0.0f;

[Desc("固定生命")]

public int BaseHp = 0;

[Desc("固定谋略")]

public float BaseInte = 0.0f;

[Desc("固定攻城")]

public float BaseBreak = 0.0f;

[Desc("战斗属性")]

public Dictionary<int, float> CombatProperty = new Dictionary<int, float>();

[Desc("当前兵力")]

public int CurrentHp = 0;

[Desc("战斗力")]

public int CombatPower = 0;

[Desc("武将已装备列表<装备位置, 信息>")]

public Dictionary<int, CLS\_EquipInfo> DictEquiped = new Dictionary<int, CLS\_EquipInfo>();

[Desc("星级")]

public int Satr = 0;

[Desc("觉醒等级")]

public int WakeLevel = 0;

[Desc("武将觉醒列表<觉醒位置, 等级>")]

public Dictionary<int, int> DictWake = new Dictionary<int, int>();

[Desc("武将强化列表<强化id, 值>")]

public Dictionary<int, float> DictStremgthen = new Dictionary<int, float>();

### 协议类 C2G\_Warrior\_WarriorList 请求 武将列表

对应ID： C2G\_WARRIOR\_WARRIORLIST

### 协议类 G2C\_Warrior\_WarriorList 返回 武将列表

对应ID： G2C\_WARRIOR\_WARRIORLIST

[Desc("武将列表")]

public List<CLS\_WarriorInfo> ListWarrior = new List<CLS\_WarriorInfo>();

[Desc("上阵列表")]

public List<long> ListInBattle = new List<long>();

### 协议类 C2G\_Warrior\_Lock 请求 武将锁定/解锁

对应ID： C2G\_WARRIOR\_LOCK

[Desc("唯一ID")]

public long Id = 0;

### 协议类 G2C\_Warrior\_Lock 返回 武将锁定/解锁

对应ID： G2C\_WARRIOR\_LOCK

[Desc("唯一ID")]

public long Id = 0;

### 协议类 C2G\_Warrior\_Advance 请求 武将进阶

对应ID： C2G\_WARRIOR\_ADVANCE

[Desc("唯一ID")]

public long Id = 0;

[Desc("材料武将ID")]

public List<long> MateriaWarriorID = new List<long>();

### 协议类 G2C\_Warrior\_Advance 返回 武将进阶

对应ID： G2C\_WARRIOR\_ADVANCE

[Desc("武将信息")]

public CLS\_WarriorInfo WarriorInfo = new CLS\_WarriorInfo();

### 协议类 C2G\_Warrior\_Rename 请求 武将修改昵称

对应ID： C2G\_WARRIOR\_RENAME

[Desc("唯一ID")]

public long Id = 0;

[Desc("武将自定义名称")]

public string MarkName = "";

### 协议类 G2C\_Warrior\_Rename 返回 武将修改昵称

对应ID： G2C\_WARRIOR\_RENAME

### 协议类 C2G\_Warrior\_UpLevel 请求 武将升级

对应ID： C2G\_WARRIOR\_UPLEVEL

[Desc("唯一ID")]

public long Id = 0;

[Desc("使用武将信息")]

public List<long> ListWarrior = new List<long>();

### 协议类 G2C\_Warrior\_UpLevel 返回 武将升级

对应ID： G2C\_WARRIOR\_UPLEVEL

[Desc("唯一ID")]

public long Id = 0;

[Desc("等级")]

public int Level = 0;

[Desc("经验")]

public int Exp = 0;

### 协议类 G2C\_Warrior\_NotifyLevelUp 通知 武将升级

对应ID： G2C\_WARRIOR\_NOTIFYLEVELUP

### 协议类 C2G\_Warrior\_ImproveInfo 请求 武将培养信息

对应ID： C2G\_WARRIOR\_IMPROVEINFO

### 协议类 G2C\_Warrior\_ImproveInfo 返回 武将培养信息

对应ID： G2C\_WARRIOR\_IMPROVEINFO

[Desc("道具列表(武将培养类)")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

### 协议类 C2G\_Warrior\_Improve 请求 武将培养

对应ID： C2G\_WARRIOR\_IMPROVE

[Desc("唯一ID")]

public long Id = 0;

[Desc("使用道具信息")]

public CLS\_ItemInfo ItemInfo = new CLS\_ItemInfo();

### 协议类 G2C\_Warrior\_Improve 返回 武将培养

对应ID： G2C\_WARRIOR\_IMPROVE

### 协议类 C2G\_Warrior\_Star 请求 武将升星

对应ID： C2G\_WARRIOR\_STAR

[Desc("升星武将唯一ID")]

public long Id = 0;

[Desc("材料武将ID")]

public List<long> MateriaWarriorID = new List<long>();

### 协议类 G2C\_Warrior\_Star 返回 武将升星

对应ID： G2C\_WARRIOR\_STAR

### 协议类 C2G\_Warrior\_Skill 请求 武将技能

对应ID： C2G\_WARRIOR\_SKILL

[Desc("武将唯一ID")]

public long Id = 0;

[Desc("技能ID")]

public int Skill = 0;

### 协议类 G2C\_Warrior\_Skill 返回 武将技能

对应ID： G2C\_WARRIOR\_SKILL

### 协议类 C2G\_Warrior\_WakeUnlock 请求 武将觉醒条件激活

对应ID： C2G\_WARRIOR\_WAKEUNLOCK

[Desc("武将唯一ID")]

public long Id = 0;

[Desc("激活类型 EWakeDataType")]

public int Type = 0;

### 协议类 G2C\_Warrior\_WakeUnlock 返回 武将觉醒条件激活

对应ID： G2C\_WARRIOR\_WAKEUNLOCK

### 协议类 C2G\_Warrior\_WakeInfo 请求 武将觉醒材料信息

对应ID： C2G\_WARRIOR\_WAKEINFO

[Desc("武将唯一ID")]

public long Id = 0;

[Desc("激活类型 EWakeDataType")]

public int Type = 0;

### 协议类 G2C\_Warrior\_WakeInfo 返回 武将觉醒材料信息

对应ID： G2C\_WARRIOR\_WAKEINFO

[Desc("需求道具列表")]

public List<CLS\_ItemNeedInfo> ListItemNeedInfo = new List<CLS\_ItemNeedInfo>();

[Desc("最大关卡ID")]

public int maxStoryId = 0;

### 协议类 C2G\_Warrior\_WakeUp 请求 武将觉醒

对应ID： C2G\_WARRIOR\_WAKEUP

[Desc("唯一ID")]

public long Id = 0;

### 协议类 G2C\_Warrior\_WakeUp 返回 武将觉醒

对应ID： G2C\_WARRIOR\_WAKEUP

### 协议类 C2G\_Warrior\_Sell 请求 武将出售

对应ID： C2G\_WARRIOR\_SELL

[Desc("唯一ID")]

public List<long> ListIds = new List<long>();

### 协议类 G2C\_Warrior\_Sell 返回 武将出售

对应ID： G2C\_WARRIOR\_SELL

[Desc("卖店价格")]

public int Money = 0;

## 封地

### 公共类 CLS\_BuildingPlayerInfo 封地建筑信息

[Desc("ID(位置)")]

public int Id = 0;

[Desc("建筑类型")]

public int BuildingType = 0;

[Desc("等级")]

public int Level = 0;

[Desc("容量")]

public long Capacity = 0;

[Desc("库存")]

public long Stock = 0;

[Desc("仓库容量")]

public long StoreCapacity = 0;

[Desc("仓库库存")]

public long StoreStock = 0;

[Desc("容量下一级增量")]

public long CapacityNextLevelAdd = 0;

[Desc("是否在建造中")]

public bool IsInBuilding = false;

[Desc("剩余毫秒数")]

public long BuildEndMs = 0;

[Desc("每小时产量")]

public int YieldPerHour = 0;

[Desc("每小时产量下一级增量")]

public int YieldPerHourNextLevelAdd = 0;

[Desc("建造消耗时间")]

public TimeSpan CostTime = new TimeSpan();

[Desc("建造消耗游戏币")]

public int CostGold = 0;

[Desc("建造消耗粮草")]

public int CostFood = 0;

[Desc("建造消耗木材")]

public int CostWood = 0;

[Desc("建造消耗铁矿")]

public int CostIron = 0;

### 公共类 CLS\_BuildingPlayerInfoSimple 封地建筑简易信息 用来推送

[Desc("ID(位置)")]

public int Id = 0;

[Desc("库存")]

public long Stock = 0;

### 协议类 C2G\_BuildingPlayer\_List 请求 封地建筑列表

对应ID： C2G\_BUILDINGPLAYER\_LIST

### 协议类 G2C\_BuildingPlayer\_List 返回 封地建筑列表

对应ID： G2C\_BUILDINGPLAYER\_LIST

[Desc("封地建筑列表")]

public Dictionary<int, CLS\_BuildingPlayerInfo> DictBuilding = new Dictionary<int, CLS\_BuildingPlayerInfo>();

### 协议类 G2C\_BuildingPlayer\_PushInfoSimple 推送 封地建筑简易信息

对应ID： G2C\_BUILDINGPLAYER\_PUSHINFOSIMPLE

[Desc("封地建筑简易信息")]

public Dictionary<int, CLS\_BuildingPlayerInfoSimple> DictBuilding = new Dictionary<int, CLS\_BuildingPlayerInfoSimple>();

### 协议类 C2G\_BuildingPlayer\_Harvest 请求 封地建筑收获

对应ID： C2G\_BUILDINGPLAYER\_HARVEST

[Desc("ID(位置)")]

public int Id = 0;

### 协议类 G2C\_BuildingPlayer\_Harvest 返回 封地建筑收获

对应ID： G2C\_BUILDINGPLAYER\_HARVEST

[Desc("收取数量")]

public int CountHarvest = 0;

[Desc("刷新建筑信息")]

public CLS\_BuildingPlayerInfo Building = new CLS\_BuildingPlayerInfo();

### 协议类 C2G\_BuildingPlayer\_HarvestAll 请求 封地建筑一键收获

对应ID： C2G\_BUILDINGPLAYER\_HARVESTALL

### 协议类 G2C\_BuildingPlayer\_HarvestAll 返回 封地建筑一键收获

对应ID： G2C\_BUILDINGPLAYER\_HARVESTALL

[Desc("封地建筑列表")]

public Dictionary<int, CLS\_BuildingPlayerInfo> DictBuilding = new Dictionary<int, CLS\_BuildingPlayerInfo>();

### 协议类 C2G\_BuildingPlayer\_Build 请求 封地建筑建造/升级

对应ID： C2G\_BUILDINGPLAYER\_BUILD

[Desc("ID(位置)")]

public int Id = 0;

### 协议类 G2C\_BuildingPlayer\_Build 返回 封地建筑建造/升级

对应ID： G2C\_BUILDINGPLAYER\_BUILD

[Desc("刷新建筑信息")]

public CLS\_BuildingPlayerInfo Building = new CLS\_BuildingPlayerInfo();

### 协议类 G2C\_BuildingPlayer\_NotifyBuilt 通知建筑完成

对应ID： G2C\_BUILDINGPLAYER\_NOTIFYBUILT

[Desc("刷新建筑信息")]

public CLS\_BuildingPlayerInfo Building = new CLS\_BuildingPlayerInfo();

### 公共类 CLS\_SmithSimple 匠工坊单条信息 用来推送

[Desc("ID")]

public int Id = 0;

[Desc("需求道具列表")]

public List<CLS\_ItemNeedInfo> ListItemNeedInfo = new List<CLS\_ItemNeedInfo>();

[Desc("是否解锁")]

public bool Lock = false;

[Desc("已有数量")]

public int num = 0;

### 协议类 C2G\_Smith\_Start 请求 匠工坊列表

对应ID： C2G\_SMITH\_START

### 协议类 G2C\_Smith\_Start 返回 匠工坊列表

对应ID： G2C\_SMITH\_START

[Desc("唯一ID列表 匠工房制作消耗")]

public Dictionary<int, CLS\_SmithSimple> DicSmithInfo = new Dictionary<int, CLS\_SmithSimple>();

### 协议类 C2G\_Smith\_Do 请求 匠工坊制作

对应ID： C2G\_SMITH\_DO

[Desc("唯一ID")]

public int ID = 0;

[Desc("制作数量")]

public int Count = 0;

### 协议类 G2C\_Smith\_Do 返回 匠工坊制作 产出

对应ID： G2C\_SMITH\_DO

[Desc("匠工房产出")]

public List<CLS\_AwardItem> Items = new List<CLS\_AwardItem>();

[Desc("装备信息")]

public List<CLS\_EquipInfo> EquipInfo = new List<CLS\_EquipInfo>();

### 公共类 CLS\_ScienceInfo 兵法信息

[Desc("兵法类型")]

public int ScienceType = 0;

[Desc("等级")]

public int Level = 0;

[Desc("配置ID")]

public int ConfigID = 0;

[Desc("是否在建造中")]

public bool IsInLearning = false;

[Desc("剩余毫秒数")]

public long LearnEndMs = 0;

[Desc("立即完成需要金币")]

public long CompleteNeed = 0;

### 协议类 C2G\_Science\_List 请求 兵法列表

对应ID： C2G\_SCIENCE\_LIST

### 协议类 G2C\_Science\_List 返回 兵法列表

对应ID： G2C\_SCIENCE\_LIST

[Desc("兵法列表")]

public Dictionary<int, CLS\_ScienceInfo> DictScience = new Dictionary<int, CLS\_ScienceInfo>();

### 协议类 C2G\_Science\_Learn 请求 兵法研究

对应ID： C2G\_SCIENCE\_LEARN

[Desc("ID(位置)")]

public int Id = 0;

### 协议类 G2C\_Science\_Learn 返回 兵法研究

对应ID： G2C\_SCIENCE\_LEARN

[Desc("兵法信息")]

public CLS\_ScienceInfo Info = new CLS\_ScienceInfo();

### 协议类 C2G\_Science\_Info 请求 兵法信息

对应ID： C2G\_SCIENCE\_INFO

[Desc("ID(位置)")]

public int Id = 0;

### 协议类 G2C\_Science\_Info 返回 兵法信息

对应ID： G2C\_SCIENCE\_INFO

[Desc("兵法信息")]

public CLS\_ScienceInfo Info = new CLS\_ScienceInfo();

### 协议类 C2G\_Science\_Cancel 请求 兵法取消研究

对应ID： C2G\_SCIENCE\_CANCEL

[Desc("ID(位置)")]

public int Id = 0;

### 协议类 G2C\_Science\_Cancel 返回 兵法取消研究

对应ID： G2C\_SCIENCE\_CANCEL

[Desc("兵法信息")]

public CLS\_ScienceInfo Info = new CLS\_ScienceInfo();

### 协议类 G2C\_Science\_Notify 通知兵法完成

对应ID： G2C\_SCIENCE\_NOTIFY

[Desc("刷新兵法信息")]

public CLS\_ScienceInfo ScienceInfo = new CLS\_ScienceInfo();

### 公共类 CLS\_RecruitInfo 兵营募兵信息

[Desc("ID(位置)")]

public int Id = 0;

[Desc("募兵状态 ERecruitStatus")]

public byte ERecruitStatus = 0;

[Desc("募兵开始时间")]

public DateTime DtRecruitStart = DateTime.Now;

[Desc("募兵结束时间")]

public DateTime DtRecruitEnd = DateTime.Now;

### 协议类 C2G\_Recruit\_Info 请求 兵营募兵信息

对应ID： C2G\_RECRUIT\_INFO

[Desc("ID(位置)")]

public int Id = 0;

### 协议类 G2C\_Recruit\_Info 返回 兵营募兵信息

对应ID： G2C\_RECRUIT\_INFO

[Desc("ID(位置)")]

public int Id = 0;

[Desc("兵营募兵信息")]

public CLS\_RecruitInfo RecruitInfo = new CLS\_RecruitInfo();

### 协议类 C2G\_Recruit\_Start 请求 兵营募兵

对应ID： C2G\_RECRUIT\_START

[Desc("ID(位置)")]

public int Id = 0;

[Desc("数量")]

public int Count = 0;

### 协议类 G2C\_Recruit\_Start 返回 兵营募兵

对应ID： G2C\_RECRUIT\_START

[Desc("ID(位置)")]

public int Id = 0;

[Desc("数量")]

public int Count = 0;

[Desc("兵营募兵信息")]

public CLS\_RecruitInfo RecruitInfo = new CLS\_RecruitInfo();

### 协议类 G2C\_Recruit\_NotifyComplete 通知募兵完成

对应ID： G2C\_RECRUIT\_NOTIFYCOMPLETE

[Desc("刷新建筑信息")]

public CLS\_BuildingPlayerInfo Building = new CLS\_BuildingPlayerInfo();

### 协议类 C2G\_Player\_SpeedupInfo 请求 通用加速信息

对应ID： C2G\_PLAYER\_SPEEDUPINFO

[Desc("ID(位置)")]

public int Id = 0;

[Desc("类型ESpeedup")]

public int Type = 0;

### 协议类 G2C\_Player\_SpeedupInfo 返回 通用加速信息

对应ID： G2C\_PLAYER\_SPEEDUPINFO

[Desc("剩余毫秒数")]

public long EndMs = 0;

[Desc("立即完成需要金币")]

public long CompleteNeed = 0;

[Desc("道具列表(加速类)")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

### 协议类 C2G\_BuildingPlayer\_SpeedupComplete 请求 封地建筑建造加速立即完成

对应ID： C2G\_BUILDINGPLAYER\_SPEEDUPCOMPLETE

[Desc("ID(位置)")]

public int Id = 0;

### 协议类 G2C\_BuildingPlayer\_SpeedupComplete 返回 封地建筑建造加速立即完成

对应ID： G2C\_BUILDINGPLAYER\_SPEEDUPCOMPLETE

[Desc("刷新建筑信息")]

public CLS\_BuildingPlayerInfo Building = new CLS\_BuildingPlayerInfo();

### 协议类 C2G\_Recruit\_Speedup 请求 兵营募兵加速

对应ID： C2G\_RECRUIT\_SPEEDUP

[Desc("ID(位置)")]

public int Id = 0;

[Desc("类型ESpeedup")]

public int Type = 0;

### 协议类 G2C\_Recruit\_Speedup 返回 兵营募兵加速

对应ID： G2C\_RECRUIT\_SPEEDUP

[Desc("ID(位置)")]

public int Id = 0;

[Desc("兵营募兵信息")]

public CLS\_RecruitInfo RecruitInfo = new CLS\_RecruitInfo();

### 协议类 C2G\_Science\_Fast 请求 兵法立即研究或加速信息

对应ID： C2G\_SCIENCE\_FAST

[Desc("ID(位置)")]

public int Id = 0;

### 协议类 G2C\_Science\_Fast 返回 兵法立即研究或加速信息

对应ID： G2C\_SCIENCE\_FAST

[Desc("兵法信息")]

public CLS\_ScienceInfo Info = new CLS\_ScienceInfo();

### 协议类 C2G\_Player\_SpeedupBuy 请求 加速道具购买

对应ID： C2G\_PLAYER\_SPEEDUPBUY

[Desc("ID ShopSpeedUp表里的ID")]

public int Id = 0;

[Desc("数量")]

public int Amount = 0;

[Desc("类型ESpeedup")]

public int Type = 0;

### 协议类 G2C\_Player\_SpeedupBuy 返回 加速道具购买

对应ID： G2C\_PLAYER\_SPEEDUPBUY

[Desc("道具列表(加速类)")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

### 协议类 C2G\_Player\_Speedup 请求 加速使用道具

对应ID： C2G\_PLAYER\_SPEEDUP

[Desc("ID(位置)")]

public int Id = 0;

[Desc("道具")]

public List<CLS\_ItemInfo> Items = new List<CLS\_ItemInfo>();

[Desc("类型ESpeedup")]

public int Type = 0;

### 协议类 G2C\_Player\_Speedup 返回 加速使用道具

对应ID： G2C\_PLAYER\_SPEEDUP

[Desc("类型ESpeedup")]

public int Type = 0;

[Desc("刷新建筑信息")]

public CLS\_BuildingPlayerInfo Building = new CLS\_BuildingPlayerInfo();

[Desc("兵营募兵信息")]

public CLS\_RecruitInfo RecruitInfo = new CLS\_RecruitInfo();

[Desc("兵法信息")]

public CLS\_ScienceInfo ScienceInfo = new CLS\_ScienceInfo();

[Desc("道具列表(加速类)")]

public List<CLS\_ItemInfo> ListItem = new List<CLS\_ItemInfo>();

[Desc("剩余毫秒数")]

public long EndMs = 0;

[Desc("立即完成需要金币")]

public long CompleteNeed = 0;

### 协议类 C2G\_Help\_Fast 请求 小助手

对应ID： C2G\_HELP\_FAST

### 协议类 G2C\_Help\_Fast 返回 小助手

对应ID： G2C\_HELP\_FAST

[Desc("建筑队列1信息")]

public CLS\_BuildingPlayerInfo Building1 = new CLS\_BuildingPlayerInfo();

[Desc("建筑队列2信息")]

public CLS\_BuildingPlayerInfo Building2 = new CLS\_BuildingPlayerInfo();

[Desc("兵法信息")]

public CLS\_ScienceInfo ScienceInfo = new CLS\_ScienceInfo();

[Desc("步兵营募兵信息")]

public CLS\_RecruitInfo RecruitInfoBu = new CLS\_RecruitInfo();

[Desc("骑兵营募兵信息")]

public CLS\_RecruitInfo RecruitInfoQi = new CLS\_RecruitInfo();

[Desc("弓兵营募兵信息")]

public CLS\_RecruitInfo RecruitInfoGong = new CLS\_RecruitInfo();

## 城池

### 公共类 CLS\_CityInfoBase 城池信息

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("太守Id")]

public long LeaderPuid = 0;

[Desc("太守名字")]

public string LeaderName = "";

[Desc("繁荣度")]

public long Prosperity = 0;

### 公共类 CLS\_CityInfo4Guild 势力城池信息

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("太守Id")]

public long LeaderPuid = 0;

[Desc("太守名字")]

public string LeaderName = "";

[Desc("繁荣度")]

public long Prosperity = 0;

[Desc("本周收入(府库库存)")]

public int Revenue = 0;

### 协议类 C2G\_City\_CityInfo 请求 城池信息

对应ID： C2G\_CITY\_CITYINFO

[Desc("城池唯一ID(配置ID)")]

public int Uid = 0;

### 协议类 G2C\_City\_CityInfo 返回 城池信息

对应ID： G2C\_CITY\_CITYINFO

[Desc("城池信息")]

public CLS\_CityInfoBase CityInfo = new CLS\_CityInfoBase();

### 协议类 C2G\_City\_ListCity 请求 城池列表

对应ID： C2G\_CITY\_LISTCITY

### 协议类 G2C\_City\_ListCity 返回 城池列表

对应ID： G2C\_CITY\_LISTCITY

[Desc("都城ID")]

public int CapitalCity = 0;

[Desc("城池列表")]

public Dictionary<int, CLS\_CityInfoBase> DictCity = new Dictionary<int, CLS\_CityInfoBase>();

### 协议类 C2G\_City\_DepotInfo 请求 府库

对应ID： C2G\_CITY\_DEPOTINFO

[Desc("城池ID")]

public int CityUid = 0;

### 协议类 G2C\_City\_DepotInfo 返回 府库

对应ID： G2C\_CITY\_DEPOTINFO

[Desc("繁荣度")]

public long Prosperity = 0;

[Desc("本日预期收益")]

public int RevenueDay = 0;

[Desc("本期预期收益")]

public int RevenueAll = 0;

## 城池建筑

### 公共类 CLS\_BuildingItemNeedInfo 建筑道具需求

[Desc("道具Id")]

public int ItemId = 0;

[Desc("已建")]

public int Built = 0;

[Desc("需求")]

public int Need = 0;

[Desc("自己拥有")]

public int Value = 0;

### 公共类 CLS\_CityBuildingInfo 城池建筑信息

[Desc("(位置)")]

public int Pos = 0;

[Desc("建筑类型")]

public int BuildingType = 0;

[Desc("等级")]

public int Level = 0;

[Desc("建筑道具需求")]

public Dictionary<int, CLS\_BuildingItemNeedInfo> DictNeedInfo = new Dictionary<int, CLS\_BuildingItemNeedInfo>();

### 协议类 C2G\_CityBuilding\_List 请求 建筑列表

对应ID： C2G\_CITYBUILDING\_LIST

[Desc("城池ID")]

public int Uid = 0;

### 协议类 G2C\_CityBuilding\_List 返回 建筑列表

对应ID： G2C\_CITYBUILDING\_LIST

[Desc("府城建筑列表")]

public Dictionary<int, CLS\_CityBuildingInfo> DictBuilding = new Dictionary<int, CLS\_CityBuildingInfo>();

### 协议类 C2G\_CityBuilding\_Build 请求 建筑升级

对应ID： C2G\_CITYBUILDING\_BUILD

[Desc("城池ID")]

public int Uid = 0;

[Desc("(位置)")]

public int Pos = 0;

[Desc("建筑使用(道具ID 数量)")]

public Dictionary<int, int> DictBuildUse = new Dictionary<int, int>();

### 协议类 G2C\_CityBuilding\_Build 返回 建筑升级

对应ID： G2C\_CITYBUILDING\_BUILD

[Desc("建筑信息")]

public CLS\_CityBuildingInfo BuildingInfo = new CLS\_CityBuildingInfo();

## 城池城墙

### 公共类 CLS\_CampWallInfo 守备营城墙信息

[Desc("唯一ID(1-5)(1=主营)")]

public int Uid = 0;

[Desc("城墙耐久")]

public CLS\_CountInfo WallHp = new CLS\_CountInfo();

[Desc("可修复CD")]

public TimeSpan TsCooldownRepair = new TimeSpan();

[Desc("滚木")]

public int Bowls = 0;

[Desc("拒马")]

public int Frise = 0;

[Desc("陷阱")]

public int Trap = 0;

[Desc("建筑道具需求")]

public Dictionary<int, CLS\_ItemNeedInfo> DictNeedInfo = new Dictionary<int, CLS\_ItemNeedInfo>();

### 公共类 CLS\_WallInfo 城墙信息

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("等级")]

public int Level = 0;

[Desc("守备营城墙信息")]

public Dictionary<int, CLS\_CampWallInfo> DictCampWall = new Dictionary<int, CLS\_CampWallInfo>();

### 协议类 C2G\_Wall\_Info 请求 城墙信息

对应ID： C2G\_WALL\_INFO

[Desc("城池ID")]

public int CityUid = 0;

### 协议类 G2C\_Wall\_Info 返回 城墙信息

对应ID： G2C\_WALL\_INFO

[Desc("城池ID")]

public int CityUid = 0;

[Desc("城墙信息")]

public CLS\_WallInfo WallInfo = new CLS\_WallInfo();

### 协议类 C2G\_Wall\_Repair 请求 城墙修复

对应ID： C2G\_WALL\_REPAIR

[Desc("城池ID")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

### 协议类 G2C\_Wall\_Repair 返回 城墙修复

对应ID： G2C\_WALL\_REPAIR

[Desc("城池ID")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("城墙信息")]

public CLS\_WallInfo WallInfo = new CLS\_WallInfo();

### 协议类 C2G\_Wall\_MakeWork 请求 城墙建造器械

对应ID： C2G\_WALL\_MAKEWORK

[Desc("城池ID")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("城墙器械类型 EWallWorkType")]

public int WallWorkType = 0;

[Desc("建造数目")]

public int MakeCount = 0;

### 协议类 G2C\_Wall\_MakeWork 返回 城墙建造器械

对应ID： G2C\_WALL\_MAKEWORK

[Desc("城池ID")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("城墙器械类型 EWallWorkType")]

public int WallWorkType = 0;

[Desc("建造数目")]

public int MakeCount = 0;

[Desc("城墙信息")]

public CLS\_WallInfo WallInfo = new CLS\_WallInfo();

## 城池守备营

### 公共类 CLS\_CampNpcInfo 守军NPC信息

[Desc("唯一ID(1-n)")]

public int Uid = 0;

[Desc("关卡配置ID")]

public int StageId = 0;

[Desc("存活")]

public bool IsAlive = false;

[Desc("剩余复活时间")]

public TimeSpan TsRevive = new TimeSpan();

### 公共类 CLS\_CampInfo 守备营信息

[Desc("唯一ID(1-5)(1=主营)")]

public int Uid = 0;

[Desc("守军NPC")]

public Dictionary<int, CLS\_CampNpcInfo> DictCampNpc = new Dictionary<int, CLS\_CampNpcInfo>();

### 协议类 C2G\_Camp\_Info 请求 守备营信息

对应ID： C2G\_CAMP\_INFO

[Desc("城池ID")]

public int CityUid = 0;

### 协议类 G2C\_Camp\_Info 返回 守备营信息

对应ID： G2C\_CAMP\_INFO

[Desc("城池ID")]

public int CityUid = 0;

[Desc("守备营信息")]

public Dictionary<int, CLS\_CampInfo> DictCamp = new Dictionary<int, CLS\_CampInfo>();

## 城池珍宝阁

### 协议类 C2G\_City\_ShopInfo 请求 城池珍宝阁信息

对应ID： C2G\_CITY\_SHOPINFO

[Desc("城池ID")]

public int CityUid = 0;

### 协议类 G2C\_City\_ShopInfo 返回 城池珍宝阁信息

对应ID： G2C\_CITY\_SHOPINFO

[Desc("城池ID")]

public int CityUid = 0;

[Desc("商品信息")]

public List<CLS\_ShopGoodsInfo> GoodsInfo = new List<CLS\_ShopGoodsInfo>();

[Desc("特产商品信息")]

public Dictionary<int, int> SpecialtyGoodsInfo = new Dictionary<int, int>();

[Desc("拥有的贡献")]

public int Contribution = 0;

### 协议类 C2G\_City\_ShopBuy 请求 处理城池珍宝阁购买

对应ID： C2G\_CITY\_SHOPBUY

[Desc("城池ID")]

public int CityUid = 0;

[Desc("配置ID")]

public int Config = 0;

[Desc("数量")]

public int Amount = 0;

### 协议类 G2C\_City\_ShopBuy 返回 处理城池珍宝阁购买

对应ID： G2C\_CITY\_SHOPBUY

[Desc("商品信息")]

public List<CLS\_ShopGoodsInfo> GoodsInfo = new List<CLS\_ShopGoodsInfo>();

[Desc("拥有的贡献")]

public int Contribution = 0;

[Desc("城池ID")]

public int CityUid = 0;

[Desc("特产商品信息")]

public Dictionary<int, int> SpecialtyGoodsInfo = new Dictionary<int, int>();

## 商城

### 公共类 CLS\_ShopPageInfo 商城页面信息

[Desc("ID")]

public int Id = 0;

[Desc("商品列表")]

public Dictionary<int, CLS\_ShopGoodsInfo> DictGoods = new Dictionary<int, CLS\_ShopGoodsInfo>();

[Desc("剩余刷新时间 -1=无限制")]

public int Cooldown = 0;

### 公共类 CLS\_ShopGoodsInfo 商城货物信息

[Desc("ID")]

public int Id = 0;

[Desc("已有数量")]

public int Count = 0;

[Desc("可买数量 -1=无限制")]

public int AmoutCanBuy = 0;

### 协议类 C2G\_Shop\_List 请求 商城列表

对应ID： C2G\_SHOP\_LIST

### 协议类 G2C\_Shop\_List 返回 商城列表

对应ID： G2C\_SHOP\_LIST

[Desc("商城页面")]

public Dictionary<int, CLS\_ShopPageInfo> DictPage = new Dictionary<int, CLS\_ShopPageInfo>();

[Desc("钻石商城累计刷新次数")]

public int RefreshTotal = 0;

[Desc("钻石商城当前刷新消耗")]

public int RefreshCost = 0;

### 协议类 C2G\_Shop\_Buy 请求 商城购买

对应ID： C2G\_SHOP\_BUY

[Desc("ID")]

public int Id = 0;

[Desc("购买数量")]

public int Amount = 0;

### 协议类 G2C\_Shop\_Buy 返回 商城购买

对应ID： G2C\_SHOP\_BUY

[Desc("商城页面")]

public Dictionary<int, CLS\_ShopPageInfo> DictPage = new Dictionary<int, CLS\_ShopPageInfo>();

### 协议类 C2G\_Shop\_QuickBuy 请求 商城快捷购买

对应ID： C2G\_SHOP\_QUICKBUY

[Desc("ID")]

public int Id = 0;

[Desc("购买数量")]

public int Amount = 0;

### 协议类 G2C\_Shop\_QuickBuy 返回 商城快捷购买

对应ID： G2C\_SHOP\_QUICKBUY

[Desc("ID")]

public int Id = 0;

### 协议类 C2G\_Refresh\_Diamond\_Store 请求 刷新钻石商城

对应ID： C2G\_REFRESH\_DIAMOND\_STORE

### 协议类 G2C\_Refresh\_Diamond\_Store 返回 刷新钻石商城

对应ID： G2C\_REFRESH\_DIAMOND\_STORE

[Desc("累计刷新次数")]

public int RefreshTotal = 0;

[Desc("当前刷新消耗")]

public int RefreshCost = 0;

[Desc("商城页面")]

public Dictionary<int, CLS\_ShopPageInfo> DictPage = new Dictionary<int, CLS\_ShopPageInfo>();

### 协议类 C2G\_Shop\_BuyWarriorBag 请求 商城购买武将背包上限

对应ID： C2G\_SHOP\_BUYWARRIORBAG

[Desc("ID")]

public int Id = 0;

### 协议类 G2C\_Shop\_BuyWarriorBag 返回 商城购买武将背包上限

对应ID： G2C\_SHOP\_BUYWARRIORBAG

[Desc("ID")]

public int Id = 0;

### 协议类 C2G\_Buy\_Gold 请求 金币兑换

对应ID： C2G\_BUY\_GOLD

public int times = 0;

### 协议类 G2C\_Buy\_Gold 返回

对应ID： G2C\_BUY\_GOLD

[Desc("增加的金币")]

public int addGold = 0;

[Desc("消耗的元宝")]

public int useGem = 0;

## 交易所

### 公共类 CLS\_DesignInfo 单子信息

[Desc("单号 唯一ID")]

public long Id = 0;

[Desc("交易方式 对应EDesignMode")]

public int DesignMode = 0;

[Desc("挂单玩家唯一ID")]

public long Puid = 0;

[Desc("挂单玩家名字")]

public string Name = "";

[Desc("单手价格(每手100)")]

public long Price = 0;

[Desc("挂单数量")]

public long Count = 0;

[Desc("到期时间")]

public DateTime DtExpire = DateTime.Now;

[Desc("剩余时间")]

public TimeSpan TsRemainder = new TimeSpan();

### 公共类 CLS\_TransactionInfo 上次交易信息

[Desc("成交单价(每手100)")]

public long Price = 0;

[Desc("交易时间")]

public DateTime DtTransaction = DateTime.Now;

### 协议类 C2G\_Exchange\_List 请求 挂单列表

对应ID： C2G\_EXCHANGE\_LIST

[Desc("交易类型 对应EDesignFlag")]

public int DesignFlag = 0;

[Desc("交易方式 对应EDesignMode")]

public int DesignMode = 0;

[Desc("页码1-n")]

public int PageIndex = 0;

### 协议类 G2C\_Exchange\_List 返回 挂单列表

对应ID： G2C\_EXCHANGE\_LIST

[Desc("交易类型 对应EDesignFlag")]

public int DesignFlag = 0;

[Desc("交易方式 对应EDesignMode")]

public int DesignMode = 0;

[Desc("页码1-n")]

public int PageIndex = 0;

[Desc("单子总数目")]

public int DesignCount = 0;

[Desc("挂单列表")]

public List<CLS\_DesignInfo> ListDesign = new List<CLS\_DesignInfo>();

[Desc("上次交易信息(Price=0时未发生交易)")]

public CLS\_TransactionInfo TransactionInfo = new CLS\_TransactionInfo();

### 协议类 C2G\_Exchange\_Shelves 请求 挂单买卖

对应ID： C2G\_EXCHANGE\_SHELVES

[Desc("交易方式 对应EDesignMode")]

public int DesignMode = 0;

[Desc("单手价格(每手100)")]

public long Price = 0;

[Desc("挂单数量")]

public long Count = 0;

### 协议类 G2C\_Exchange\_Shelves 返回 挂单买卖

对应ID： G2C\_EXCHANGE\_SHELVES

[Desc("交易方式 对应EDesignMode")]

public int DesignMode = 0;

[Desc("更新单子信息")]

public CLS\_DesignInfo DesignInfo = new CLS\_DesignInfo();

### 协议类 C2G\_Exchange\_Revoke 请求 挂单撤销

对应ID： C2G\_EXCHANGE\_REVOKE

[Desc("交易方式 对应EDesignMode")]

public int DesignMode = 0;

[Desc("单号 唯一ID")]

public long Id = 0;

### 协议类 G2C\_Exchange\_Revoke 返回 挂单撤销

对应ID： G2C\_EXCHANGE\_REVOKE

### 协议类 C2G\_Exchange\_Transaction 请求 交易买卖

对应ID： C2G\_EXCHANGE\_TRANSACTION

[Desc("交易方式 对应EDesignMode")]

public int DesignMode = 0;

[Desc("单号 唯一ID")]

public long Id = 0;

[Desc("交易数量")]

public long Count = 0;

### 协议类 G2C\_Exchange\_Transaction 返回 交易买卖

对应ID： G2C\_EXCHANGE\_TRANSACTION

[Desc("交易方式 对应EDesignMode")]

public int DesignMode = 0;

[Desc("更新单子信息")]

public CLS\_DesignInfo DesignInfo = new CLS\_DesignInfo();

### 公共类 CLS\_TransactionLog 交易信息

[Desc("交易方式 对应EDesignMode")]

public int DesignMode = 0;

[Desc("货币类型")]

public int MoneyType = 0;

[Desc("交易单价")]

public long Price = 0;

[Desc("交易数量")]

public long Count = 0;

### 协议类 C2G\_GetTransactionLogList 请求 交易记录

对应ID： C2G\_GETTRANSACTIONLOGLIST

[Desc("查询类型（1:购买记录;2:出售记录）")]

public int Type = 0;

### 协议类 G2C\_GetTransactionLogList 返回 交易记录

对应ID： G2C\_GETTRANSACTIONLOGLIST

[Desc("查询类型（1:购买记录;2:出售记录）")]

public int Type = 0;

[Desc("交易记录")]

public List<CLS\_TransactionLog> ListTransactionLog = new List<CLS\_TransactionLog>();

## 商会

### 公共类 CLS\_AuctionInfoBase 商会单子信息 基础

[Desc("单号 唯一ID")]

public long Id = 0;

[Desc("挂单玩家唯一ID")]

public long Puid = 0;

[Desc("挂单玩家名字")]

public string Name = "";

[Desc("价格(单位：分)")]

public long Price = 0;

[Desc("到期时间")]

public DateTime DtExpire = DateTime.Now;

[Desc("剩余时间")]

public TimeSpan TsRemainder = new TimeSpan();

### 公共类 CLS\_AuctionInfoItem 商会单子信息 道具

[Desc("基础信息")]

public CLS\_AuctionInfoBase AuctionInfoBase = new CLS\_AuctionInfoBase();

[Desc("道具信息")]

public CLS\_ItemInfo ItemInfo = new CLS\_ItemInfo();

### 协议类 C2G\_Auction\_ListItem 请求 商会列表 道具

对应ID： C2G\_AUCTION\_LISTITEM

[Desc("排序类型 对应EOrder")]

public int Order = 0;

[Desc("交易类型 对应EAuctionFlag")]

public int AuctionFlag = 0;

[Desc("子类ID")]

public int SubType = 0;

[Desc("页码1-n (返回每页100条)")]

public int PageIndex = 0;

### 协议类 G2C\_Auction\_ListItem 返回 商会列表 道具

对应ID： G2C\_AUCTION\_LISTITEM

[Desc("排序类型 对应EOrder")]

public int Order = 0;

[Desc("交易类型 对应EAuctionFlag")]

public int AuctionFlag = 0;

[Desc("子类ID")]

public int SubType = 0;

[Desc("页码1-n (返回每页100条)")]

public int PageIndex = 0;

[Desc("总数目")]

public int TotalCount = 0;

[Desc("列表")]

public Dictionary<long, CLS\_AuctionInfoItem> DictItem = new Dictionary<long, CLS\_AuctionInfoItem>();

### 协议类 C2G\_Auction\_SellItem 请求 商会出售 道具

对应ID： C2G\_AUCTION\_SELLITEM

[Desc("道具信息")]

public CLS\_ItemInfo ItemInfo = new CLS\_ItemInfo();

[Desc("出售单价(单位：分)")]

public long Price = 0;

### 协议类 G2C\_Auction\_SellItem 返回 商会出售 道具

对应ID： G2C\_AUCTION\_SELLITEM

[Desc("商会单子信息")]

public CLS\_AuctionInfoItem AuctionInfoItem = new CLS\_AuctionInfoItem();

[Desc("刷新道具信息")]

public CLS\_ItemInfo ItemInfo = new CLS\_ItemInfo();

### 协议类 C2G\_Auction\_ReturnItem 请求 商会撤卖 道具

对应ID： C2G\_AUCTION\_RETURNITEM

[Desc("单号 唯一ID")]

public long Id = 0;

### 协议类 G2C\_Auction\_ReturnItem 返回 商会撤卖 道具

对应ID： G2C\_AUCTION\_RETURNITEM

[Desc("刷新道具信息")]

public CLS\_ItemInfo ItemInfo = new CLS\_ItemInfo();

### 协议类 C2G\_Auction\_BuyItem 请求 商会买入 道具

对应ID： C2G\_AUCTION\_BUYITEM

[Desc("单号 唯一ID")]

public long Id = 0;

### 协议类 G2C\_Auction\_BuyItem 返回 商会买入 道具

对应ID： G2C\_AUCTION\_BUYITEM

[Desc("刷新道具信息")]

public CLS\_ItemInfo ItemInfo = new CLS\_ItemInfo();

### 公共类 CLS\_AuctionInfoEquip 商会单子信息 装备

[Desc("基础信息")]

public CLS\_AuctionInfoBase AuctionInfoBase = new CLS\_AuctionInfoBase();

[Desc("装备信息")]

public CLS\_EquipInfo EquipInfo = new CLS\_EquipInfo();

### 协议类 C2G\_Auction\_ListEquip 请求 商会列表 装备

对应ID： C2G\_AUCTION\_LISTEQUIP

[Desc("排序类型 对应EOrder")]

public int Order = 0;

[Desc("交易类型 对应EAuctionFlag")]

public int AuctionFlag = 0;

[Desc("子类ID")]

public int SubType = 0;

[Desc("页码1-n (返回每页100条)")]

public int PageIndex = 0;

### 协议类 G2C\_Auction\_ListEquip 返回 商会列表 装备

对应ID： G2C\_AUCTION\_LISTEQUIP

[Desc("排序类型 对应EOrder")]

public int Order = 0;

[Desc("交易类型 对应EAuctionFlag")]

public int AuctionFlag = 0;

[Desc("子类ID")]

public int SubType = 0;

[Desc("页码1-n (返回每页100条)")]

public int PageIndex = 0;

[Desc("总数目")]

public int TotalCount = 0;

[Desc("列表")]

public Dictionary<long, CLS\_AuctionInfoEquip> DictEquip = new Dictionary<long, CLS\_AuctionInfoEquip>();

### 协议类 C2G\_Auction\_SellEquip 请求 商会出售 装备

对应ID： C2G\_AUCTION\_SELLEQUIP

[Desc("装备唯一ID")]

public long EquipId = 0;

[Desc("出售单价(单位：分)")]

public long Price = 0;

### 协议类 G2C\_Auction\_SellEquip 返回 商会出售 装备

对应ID： G2C\_AUCTION\_SELLEQUIP

[Desc("商会单子信息")]

public CLS\_AuctionInfoEquip AuctionInfoEquip = new CLS\_AuctionInfoEquip();

### 协议类 C2G\_Auction\_ReturnEquip 请求 商会撤卖 装备

对应ID： C2G\_AUCTION\_RETURNEQUIP

[Desc("单号 唯一ID")]

public long Id = 0;

### 协议类 G2C\_Auction\_ReturnEquip 返回 商会撤卖 装备

对应ID： G2C\_AUCTION\_RETURNEQUIP

[Desc("刷新装备信息")]

public CLS\_EquipInfo EquipInfo = new CLS\_EquipInfo();

### 协议类 C2G\_Auction\_BuyEquip 请求 商会买入 装备

对应ID： C2G\_AUCTION\_BUYEQUIP

[Desc("单号 唯一ID")]

public long Id = 0;

### 协议类 G2C\_Auction\_BuyEquip 返回 商会买入 装备

对应ID： G2C\_AUCTION\_BUYEQUIP

[Desc("刷新装备信息")]

public CLS\_EquipInfo EquipInfo = new CLS\_EquipInfo();

### 公共类 CLS\_AuctionInfoWarrior 商会单子信息 武将

[Desc("基础信息")]

public CLS\_AuctionInfoBase AuctionInfoBase = new CLS\_AuctionInfoBase();

[Desc("武将信息")]

public CLS\_WarriorInfo WarriorInfo = new CLS\_WarriorInfo();

### 协议类 C2G\_Auction\_ListWarrior 请求 商会列表 武将

对应ID： C2G\_AUCTION\_LISTWARRIOR

[Desc("排序类型 对应EOrder")]

public int Order = 0;

[Desc("交易类型 对应EAuctionFlag")]

public int AuctionFlag = 0;

[Desc("子类ID")]

public int SubType = 0;

[Desc("页码1-n (返回每页100条)")]

public int PageIndex = 0;

### 协议类 G2C\_Auction\_ListWarrior 返回 商会列表 武将

对应ID： G2C\_AUCTION\_LISTWARRIOR

[Desc("排序类型 对应EOrder")]

public int Order = 0;

[Desc("交易类型 对应EAuctionFlag")]

public int AuctionFlag = 0;

[Desc("子类ID")]

public int SubType = 0;

[Desc("页码1-n (返回每页100条)")]

public int PageIndex = 0;

[Desc("总数目")]

public int TotalCount = 0;

[Desc("列表")]

public Dictionary<long, CLS\_AuctionInfoWarrior> DictWarrior = new Dictionary<long, CLS\_AuctionInfoWarrior>();

### 协议类 C2G\_Auction\_SellWarrior 请求 商会出售 武将

对应ID： C2G\_AUCTION\_SELLWARRIOR

[Desc("武将唯一ID")]

public long WarriorId = 0;

[Desc("出售单价(单位：分)")]

public long Price = 0;

### 协议类 G2C\_Auction\_SellWarrior 返回 商会出售 武将

对应ID： G2C\_AUCTION\_SELLWARRIOR

[Desc("商会单子信息")]

public CLS\_AuctionInfoWarrior AuctionInfoWarrior = new CLS\_AuctionInfoWarrior();

### 协议类 C2G\_Auction\_ReturnWarrior 请求 商会撤卖 武将

对应ID： C2G\_AUCTION\_RETURNWARRIOR

[Desc("单号 唯一ID")]

public long Id = 0;

### 协议类 G2C\_Auction\_ReturnWarrior 返回 商会撤卖 武将

对应ID： G2C\_AUCTION\_RETURNWARRIOR

[Desc("刷新武将信息")]

public CLS\_WarriorInfo WarriorInfo = new CLS\_WarriorInfo();

### 协议类 C2G\_Auction\_BuyWarrior 请求 商会买入 武将

对应ID： C2G\_AUCTION\_BUYWARRIOR

[Desc("单号 唯一ID")]

public long Id = 0;

### 协议类 G2C\_Auction\_BuyWarrior 返回 商会买入 武将

对应ID： G2C\_AUCTION\_BUYWARRIOR

[Desc("刷新武将信息")]

public CLS\_WarriorInfo WarriorInfo = new CLS\_WarriorInfo();

### 协议类 C2G\_Auction\_Search 请求 商会搜索

对应ID： C2G\_AUCTION\_SEARCH

[Desc("搜索关键字")]

public string SearchKey = "";

### 协议类 G2C\_Auction\_Search 返回 商会搜索

对应ID： G2C\_AUCTION\_SEARCH

[Desc("列表")]

public Dictionary<long, CLS\_AuctionInfoItem> DictItem = new Dictionary<long, CLS\_AuctionInfoItem>();

[Desc("列表")]

public Dictionary<long, CLS\_AuctionInfoEquip> DictEquip = new Dictionary<long, CLS\_AuctionInfoEquip>();

[Desc("列表")]

public Dictionary<long, CLS\_AuctionInfoWarrior> DictWarrior = new Dictionary<long, CLS\_AuctionInfoWarrior>();

### 公共类 CLS\_AuctionLog 商会交易记录

[Desc("大类 见EAuctionLogItemType")]

public short AuctionLogItemType = 0;

[Desc("配置ID")]

public int ConfigId = 0;

[Desc("数量")]

public int Count = 0;

[Desc("交易对象名字")]

public string PlayerName = "";

[Desc("价格(单位：分)")]

public long Price = 0;

### 协议类 C2G\_Auction\_GetLog 请求 商会交易记录

对应ID： C2G\_AUCTION\_GETLOG

[Desc("查询类型 EAuctionLogType")]

public int AuctionLogType = 0;

### 协议类 G2C\_Auction\_GetLog 返回 商会交易记录

对应ID： G2C\_AUCTION\_GETLOG

[Desc("查询类型 EAuctionLogType")]

public int AuctionLogType = 0;

[Desc("交易记录")]

public List<CLS\_AuctionLog> ListAuctionLog = new List<CLS\_AuctionLog>();

## 宴会厅

### 协议类 C2G\_Draw\_GetInfo 请求 宴会厅数据

对应ID： C2G\_DRAW\_GETINFO

### 协议类 G2C\_Draw\_GetInfo 返回 宴会厅数据

对应ID： G2C\_DRAW\_GETINFO

[Desc("奖励积分")]

public int DrawScore = 0;

### 协议类 C2G\_Draw\_Luck 请求 宴会厅抽奖

对应ID： C2G\_DRAW\_LUCK

[Desc("(配置ID)")]

public int DrawPoolId = 0;

[Desc("(EDrawType)")]

public int DrawType = 0;

### 协议类 G2C\_Draw\_Luck 返回 宴会厅抽奖

对应ID： G2C\_DRAW\_LUCK

[Desc("奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Draw\_GetAward 请求 宴会厅奖励

对应ID： C2G\_DRAW\_GETAWARD

### 协议类 G2C\_Draw\_GetAward 返回 宴会厅奖励

对应ID： G2C\_DRAW\_GETAWARD

[Desc("奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

## 兑换码

### 协议类 C2G\_Cdkey\_Use 请求 使用兑换码

对应ID： C2G\_CDKEY\_USE

[Desc("兑换码")]

public string Cdkey = "";

### 协议类 G2C\_Cdkey\_Use 返回 使用兑换码

对应ID： G2C\_CDKEY\_USE

[Desc("兑换码")]

public string Cdkey = "";

[Desc("奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

## 排行榜

### 公共类 CLS\_TopData 排行榜单条信息

[Desc("排行")]

public int Rank = 0;

[Desc("唯一ID")]

public long Puid = 0;

[Desc("名字")]

public string Name = "";

[Desc("分数")]

public long Score = 0;

[Desc("存储其他信息，身份或武将名称或道具名称")]

public string Other = "";

[Desc("配置ID")]

public int ConfigId = 0;

### 协议类 C2G\_Top\_Toplist 请求 排行榜

对应ID： C2G\_TOP\_TOPLIST

[Desc("排行榜类型")]

public int type = 0;

[Desc("页码")]

public int PageIndex = 0;

### 协议类 G2C\_Top\_Toplist 返回 排行榜

对应ID： G2C\_TOP\_TOPLIST

[Desc("排行榜数据")]

public List<CLS\_TopData> TopDataList = new List<CLS\_TopData>();

[Desc("玩家数据")]

public CLS\_TopData playerTopData = new CLS\_TopData();

[Desc("排行榜类型")]

public int type = 0;

### 协议类 C2G\_Top\_Info 请求 排行榜单条具体数据信息

对应ID： C2G\_TOP\_INFO

[Desc("排行榜类型")]

public int type = 0;

[Desc("唯一ID")]

public long ID = 0;

### 协议类 G2C\_Top\_Info 返回 排行榜单条具体数据信息

对应ID： G2C\_TOP\_INFO

[Desc("排行榜类型")]

public int type = 0;

[Desc("装备信息")]

public CLS\_EquipInfo EquipInfo = new CLS\_EquipInfo();

[Desc("武将信息")]

public CLS\_WarriorInfo WarriorInfo = new CLS\_WarriorInfo();

[Desc("主角信息")]

public CLS\_PlayerData PlayerInfo = new CLS\_PlayerData();

## 任务

### 公共类 CLS\_TaskInfo 任务信息

[Desc("任务唯一ID(府衙任务用 其他任务和配置ID相同)")]

public long Tuid = 0;

[Desc("任务ID(配置ID)")]

public int Id = 0;

[Desc("任务状态 对应ETaskState")]

public byte State = 0;

[Desc("当前进度")]

public long Schedule = 0;

[Desc("最大进度")]

public long ScheduleMax = 0;

### 协议类 C2G\_Task\_List 请求 任务列表

对应ID： C2G\_TASK\_LIST

[Desc("任务类型 对应ETaskType")]

public byte TaskType = 0;

### 协议类 G2C\_Task\_List 返回 任务列表

对应ID： G2C\_TASK\_LIST

[Desc("任务类型 对应ETaskType")]

public byte TaskType = 0;

[Desc("任务列表")]

public List<CLS\_TaskInfo> ListTask = new List<CLS\_TaskInfo>();

### 协议类 C2G\_Task\_Get 请求 任务领奖

对应ID： C2G\_TASK\_GET

[Desc("任务类型 对应ETaskType")]

public byte TaskType = 0;

[Desc("任务唯一ID(府衙任务用 其他任务和配置ID相同)")]

public long Tuid = 0;

### 协议类 G2C\_Task\_Get 返回 任务领奖

对应ID： G2C\_TASK\_GET

[Desc("奖励项")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Task\_GetAll 请求 任务领奖一键领取

对应ID： C2G\_TASK\_GETALL

[Desc("任务类型 对应ETaskType")]

public byte TaskType = 0;

### 协议类 G2C\_Task\_GetAll 返回 任务领奖一键领取

对应ID： G2C\_TASK\_GETALL

[Desc("奖励项")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Task\_Everyday 请求 日常任务

对应ID： C2G\_TASK\_EVERYDAY

### 协议类 G2C\_Task\_Everyday 返回 日常任务

对应ID： G2C\_TASK\_EVERYDAY

[Desc("任务列表")]

public List<CLS\_TaskInfo> ListTask = new List<CLS\_TaskInfo>();

[Desc("活跃值")]

public int Activity = 0;

[Desc("活跃奖励领取状态(等级ID，ETaskState)")]

public Dictionary<int, int> DictTaskActivity = new Dictionary<int, int>();

### 协议类 C2G\_Task\_EverydayAward 请求 每日活跃奖励

对应ID： C2G\_TASK\_EVERYDAYAWARD

[Desc("活跃ID")]

public int ActivityID = 0;

### 协议类 G2C\_Task\_EverydayAward 返回 每日活跃奖励

对应ID： G2C\_TASK\_EVERYDAYAWARD

[Desc("奖励项")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

## 新手福利

### 公共类 CLS\_NoviceWelfare 福利信息

[Desc("当前任务ID")]

public int Id = 0;

[Desc("进度")]

public int Schedule = 0;

[Desc("(ETaskState)")]

public int State = 0;

[Desc("(EWelfareState)")]

public int endState = 0;

### 协议类 C2G\_Novice\_Welfare\_Info 请求 获取新手福利信息

对应ID： C2G\_NOVICE\_WELFARE\_INFO

### 协议类 G2C\_Novice\_Welfare\_Info 返回 获取新手福利信息

对应ID： G2C\_NOVICE\_WELFARE\_INFO

[Desc("信息列表")]

public List<CLS\_NoviceWelfare> ListInfo = new List<CLS\_NoviceWelfare>();

[Desc("所有奖励是否领取")]

public bool bHadAll = false;

### 协议类 C2G\_Novice\_Welfare\_Award 请求 领取新手福利

对应ID： C2G\_NOVICE\_WELFARE\_AWARD

[Desc("ID(配置ID)")]

public int Id = 0;

### 协议类 G2C\_Novice\_Welfare\_Award 返回 领取新手福利

对应ID： G2C\_NOVICE\_WELFARE\_AWARD

[Desc("ID(配置ID)")]

public int Id = 0;

[Desc("奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

[Desc("信息列表")]

public List<CLS\_NoviceWelfare> ListInfo = new List<CLS\_NoviceWelfare>();

[Desc("所有奖励是否领取")]

public bool bHadAll = false;

## 签到

### 公共类 CLS\_SignInfo 签到信息

[Desc("今天是否已签")]

public bool IsSigned = false;

[Desc("本轮签到总数")]

public int SignDays = 0;

[Desc("本轮补签次数")]

public int ReSignDays = 0;

[Desc("本轮天数索引")]

public int SignIndex = 0;

[Desc("剩余秒数")]

public long AliveTime = 0;

[Desc("周期秒数")]

public long TotalTime = 0;

### 协议类 C2G\_Sign\_Info 请求 获取签到信息

对应ID： C2G\_SIGN\_INFO

### 协议类 G2C\_Sign\_Info 返回 获取签到信息

对应ID： G2C\_SIGN\_INFO

[Desc("签到信息")]

public CLS\_SignInfo SignInfo = new CLS\_SignInfo();

### 协议类 C2G\_Sign 请求 处理签到

对应ID： C2G\_SIGN

### 协议类 G2C\_Sign 返回 处理签到

对应ID： G2C\_SIGN

[Desc("签到信息")]

public CLS\_SignInfo SignInfo = new CLS\_SignInfo();

[Desc("奖励信息")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_ReSign 请求 处理补签

对应ID： C2G\_RESIGN

### 协议类 G2C\_ReSign 返回 处理补签

对应ID： G2C\_RESIGN

[Desc("签到信息")]

public CLS\_SignInfo SignInfo = new CLS\_SignInfo();

[Desc("奖励信息")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

## BUFF

### 公共类 CLS\_BuffInfo BUFF信息

[Desc("配置ID")]

public int Id = 0;

[Desc("剩余秒数(-1:永久有效)")]

public long AliveTime = 0;

### 协议类 C2G\_Buff\_Info 请求 获取BUFF信息

对应ID： C2G\_BUFF\_INFO

### 协议类 G2C\_Buff\_Info 返回 获取BUFF信息

对应ID： G2C\_BUFF\_INFO

[Desc("BUFF信息")]

public List<CLS\_BuffInfo> Info = new List<CLS\_BuffInfo>();

## 活动通用

### 公共类 CLS\_AcitvityInfo 活动信息

[Desc("ID(配置ID)")]

public int Id = 0;

[Desc("进度")]

public int Schedule = 0;

[Desc("当前状态 EActivityState")]

public int State = 0;

[Desc("道具需求信息")]

public CLS\_ItemNeedInfo ItemNeedInfo = new CLS\_ItemNeedInfo();

### 协议类 C2G\_Acitvity\_Info 请求 获取活动信息

对应ID： C2G\_ACITVITY\_INFO

### 协议类 G2C\_Acitvity\_Info 返回 获取活动信息

对应ID： G2C\_ACITVITY\_INFO

[Desc("活动列表")]

public Dictionary<int, CLS\_AcitvityInfo> DictAcitvityInfo = new Dictionary<int, CLS\_AcitvityInfo>();

### 协议类 C2G\_Acitvity\_Award 请求 获取活动奖励

对应ID： C2G\_ACITVITY\_AWARD

[Desc("ID(配置ID)")]

public int Id = 0;

### 协议类 G2C\_Acitvity\_Award 返回 获取活动奖励

对应ID： G2C\_ACITVITY\_AWARD

[Desc("ID(配置ID)")]

public int Id = 0;

[Desc("奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_Acitvity\_List 请求 获取活动列表

对应ID： C2G\_ACITVITY\_LIST

### 协议类 G2C\_Acitvity\_List 返回 获取活动列表

对应ID： G2C\_ACITVITY\_LIST

public List<int> Activitys = new List<int>();

### 协议类 C2G\_Acitvity\_RebateInfo 请求 返利信息

对应ID： C2G\_ACITVITY\_REBATEINFO

### 协议类 G2C\_Acitvity\_RebateInfo 返回 返利信息

对应ID： G2C\_ACITVITY\_REBATEINFO

[Desc("本周返利")]

public long Rebate = 0;

[Desc("历史返利")]

public long RebateTotal = 0;

### 协议类 C2G\_Acitvity\_7Day 请求 获取七日试炼

对应ID： C2G\_ACITVITY\_7DAY

### 协议类 G2C\_Acitvity\_7Day 返回 获取七日试炼

对应ID： G2C\_ACITVITY\_7DAY

[Desc("任务列表")]

public List<CLS\_TaskInfo> ListTask = new List<CLS\_TaskInfo>();

### 协议类 C2G\_Acitvity\_7DayGet 请求 七日试炼领奖

对应ID： C2G\_ACITVITY\_7DAYGET

[Desc("配置ID")]

public int confID = 0;

### 协议类 G2C\_Acitvity\_7DayGet 返回 七日试炼领奖

对应ID： G2C\_ACITVITY\_7DAYGET

[Desc("奖励项")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

## 势力

### 公共类 CLS\_GuildInfoBase 势力基本信息

[Desc("唯一ID")]

public long Uid = 0;

[Desc("名字")]

public string Name = "";

[Desc("领袖名字")]

public string LeaderName = "";

[Desc("所属州")]

public int StateId = 0;

[Desc("等级")]

public int Level = 0;

[Desc("势力加入方式 EGuildJoinMode")]

public byte JoinMode = 0;

[Desc("人数")]

public CLS\_CountInfo CountMember = new CLS\_CountInfo();

[Desc("是否同盟关系")]

public bool IsAlliance = false;

### 公共类 CLS\_GuildInfoDetails 势力详细信息

[Desc("唯一ID")]

public long Uid = 0;

[Desc("名字")]

public string Name = "";

[Desc("领袖名字")]

public string LeaderName = "";

[Desc("所属州")]

public int StateId = 0;

[Desc("等级")]

public int Level = 0;

[Desc("经验")]

public int Exp = 0;

[Desc("势力资金")]

public int Fund = 0;

[Desc("总收入")]

public int Revenue = 0;

[Desc("势力加入方式 EGuildJoinMode")]

public byte JoinMode = 0;

[Desc("人数")]

public CLS\_CountInfo CountMember = new CLS\_CountInfo();

[Desc("都城ID")]

public int CapitalCity = 0;

[Desc("城池数")]

public int CountCity = 0;

[Desc("宣言")]

public string Manifesto = "";

[Desc("公告")]

public string Notice = "";

### 公共类 CLS\_GuildInfoMy 势力详细我的

[Desc("详细信息")]

public CLS\_GuildInfoDetails GuildInfoDetails = new CLS\_GuildInfoDetails();

### 协议类 C2G\_Guild\_List 请求 势力列表

对应ID： C2G\_GUILD\_LIST

### 协议类 G2C\_Guild\_List 返回 势力列表

对应ID： G2C\_GUILD\_LIST

[Desc("势力列表")]

public List<CLS\_GuildInfoBase> ListGuildInfoBase = new List<CLS\_GuildInfoBase>();

[Desc("剩余可加入时间(0为可加入)")]

public TimeSpan CdCanJoin = new TimeSpan();

### 协议类 C2G\_Guild\_ListDetails 请求 势力详情

对应ID： C2G\_GUILD\_LISTDETAILS

[Desc("唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Guild\_ListDetails 返回 势力详情

对应ID： G2C\_GUILD\_LISTDETAILS

[Desc("势力详情")]

public CLS\_GuildInfoDetails GuildInfoDetails = new CLS\_GuildInfoDetails();

### 协议类 C2G\_Guild\_Create 请求 势力创建

对应ID： C2G\_GUILD\_CREATE

[Desc("名字")]

public string Name = "";

### 协议类 G2C\_Guild\_Create 返回 势力创建

对应ID： G2C\_GUILD\_CREATE

### 协议类 C2G\_Guild\_TryLeave 请求 脱离

对应ID： C2G\_GUILD\_TRYLEAVE

### 协议类 G2C\_Guild\_TryLeave 返回 脱离

对应ID： G2C\_GUILD\_TRYLEAVE

### 协议类 C2G\_Guild\_TryKick 请求 逐出

对应ID： C2G\_GUILD\_TRYKICK

[Desc("对象玩家ID")]

public long TargetPuid = 0;

### 协议类 G2C\_Guild\_TryKick 返回 逐出

对应ID： G2C\_GUILD\_TRYKICK

### 协议类 C2G\_Guild\_Abdicate 请求 禅让

对应ID： C2G\_GUILD\_ABDICATE

[Desc("对象玩家ID")]

public long TargetPuid = 0;

### 协议类 G2C\_Guild\_Abdicate 返回 禅让

对应ID： G2C\_GUILD\_ABDICATE

### 协议类 C2G\_Guild\_TryJoin 请求 申请加入

对应ID： C2G\_GUILD\_TRYJOIN

[Desc("唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Guild\_TryJoin 返回 申请加入

对应ID： G2C\_GUILD\_TRYJOIN

### 协议类 C2G\_Guild\_ListApply 请求 申请列表

对应ID： C2G\_GUILD\_LISTAPPLY

### 协议类 G2C\_Guild\_ListApply 返回 申请列表

对应ID： G2C\_GUILD\_LISTAPPLY

[Desc("势力申请列表")]

public List<CLS\_PubPlayerBase> ListApply = new List<CLS\_PubPlayerBase>();

### 协议类 C2G\_Guild\_ApplyAction 请求 申请处理

对应ID： C2G\_GUILD\_APPLYACTION

[Desc("对象玩家ID")]

public long TargetPuid = 0;

[Desc("处理方式 EHandleAction")]

public int HandleAction = 0;

### 协议类 G2C\_Guild\_ApplyAction 返回 申请处理

对应ID： G2C\_GUILD\_APPLYACTION

[Desc("对象玩家ID")]

public long TargetPuid = 0;

[Desc("处理方式 EHandleAction")]

public int HandleAction = 0;

[Desc("人数")]

public CLS\_CountInfo CountMember = new CLS\_CountInfo();

### 公共类 CLS\_GuildMbsInfo 势力成员信息

[Desc("基本信息")]

public CLS\_PubPlayerBase PubPlayerBase = new CLS\_PubPlayerBase();

[Desc("贡献")]

public int Contribution = 0;

[Desc("历史贡献")]

public int ContributionTotal = 0;

[Desc("加入势力时间")]

public DateTime TimeJoin = DateTime.Now;

### 协议类 C2G\_Guild\_ListMbs 请求 成员列表

对应ID： C2G\_GUILD\_LISTMBS

### 协议类 G2C\_Guild\_ListMbs 返回 成员列表

对应ID： G2C\_GUILD\_LISTMBS

[Desc("势力成员列表")]

public List<CLS\_GuildMbsInfo> ListMember = new List<CLS\_GuildMbsInfo>();

### 协议类 C2G\_Guild\_ListLogRecord 请求 动态列表

对应ID： C2G\_GUILD\_LISTLOGRECORD

### 协议类 G2C\_Guild\_ListLogRecord 返回 动态列表

对应ID： G2C\_GUILD\_LISTLOGRECORD

[Desc("势力动态列表")]

public List<CLS\_GameLog> ListLog = new List<CLS\_GameLog>();

### 协议类 C2G\_Guild\_Rename 请求 改名

对应ID： C2G\_GUILD\_RENAME

[Desc("名字")]

public string Name = "";

### 协议类 G2C\_Guild\_Rename 返回 改名

对应ID： G2C\_GUILD\_RENAME

### 协议类 C2G\_Guild\_JoinMode 请求 修改加入设置

对应ID： C2G\_GUILD\_JOINMODE

[Desc("势力加入方式 EGuildJoinMode")]

public byte JoinMode = 0;

### 协议类 G2C\_Guild\_JoinMode 返回 修改加入设置

对应ID： G2C\_GUILD\_JOINMODE

### 协议类 C2G\_Guild\_Manifesto 请求 修改宣言

对应ID： C2G\_GUILD\_MANIFESTO

[Desc("宣言")]

public string Manifesto = "";

### 协议类 G2C\_Guild\_Manifesto 返回 修改宣言

对应ID： G2C\_GUILD\_MANIFESTO

### 协议类 C2G\_Guild\_Notice 请求 修改公告

对应ID： C2G\_GUILD\_NOTICE

[Desc("公告")]

public string Notice = "";

### 协议类 G2C\_Guild\_Notice 返回 修改公告

对应ID： G2C\_GUILD\_NOTICE

### 协议类 C2G\_Guild\_DonateInfo 请求 捐献信息

对应ID： C2G\_GUILD\_DONATEINFO

### 协议类 G2C\_Guild\_DonateInfo 返回 捐献信息

对应ID： G2C\_GUILD\_DONATEINFO

[Desc("捐献记录")]

public List<CLS\_GameLog> ListLog = new List<CLS\_GameLog>();

[Desc("捐献次数(key=捐献id value=今日次数)")]

public Dictionary<int, int> DictDonateCount = new Dictionary<int, int>();

[Desc("贡献值")]

public int Contribution = 0;

### 协议类 C2G\_Guild\_Donate 请求 捐献

对应ID： C2G\_GUILD\_DONATE

public int DonateId = 0;

### 协议类 G2C\_Guild\_Donate 返回 捐献

对应ID： G2C\_GUILD\_DONATE

public int DonateId = 0;

[Desc("新增捐献记录")]

public CLS\_GameLog DonateLog = new CLS\_GameLog();

[Desc("详细信息")]

public CLS\_GuildInfoDetails GuildInfoDetails = new CLS\_GuildInfoDetails();

[Desc("贡献值")]

public int Contribution = 0;

### 协议类 C2G\_Guild\_ListDiplomacy 请求 势力关系列表

对应ID： C2G\_GUILD\_LISTDIPLOMACY

### 协议类 G2C\_Guild\_ListDiplomacy 返回 势力关系列表

对应ID： G2C\_GUILD\_LISTDIPLOMACY

[Desc("势力关系列表")]

public List<CLS\_GuildInfoBase> ListGuildInfoBase = new List<CLS\_GuildInfoBase>();

### 协议类 C2G\_Guild\_SearchName 请求 势力名搜索

对应ID： C2G\_GUILD\_SEARCHNAME

[Desc("关键字")]

public string Key = "";

### 协议类 G2C\_Guild\_SearchName 返回 势力名搜索

对应ID： G2C\_GUILD\_SEARCHNAME

[Desc("势力名搜索结果")]

public List<CLS\_GuildInfoBase> ListGuildInfoBase = new List<CLS\_GuildInfoBase>();

### 协议类 C2G\_Guild\_ListAllianceApply 请求 同盟申请列表

对应ID： C2G\_GUILD\_LISTALLIANCEAPPLY

### 协议类 G2C\_Guild\_ListAllianceApply 返回 同盟申请列表

对应ID： G2C\_GUILD\_LISTALLIANCEAPPLY

[Desc("同盟申请列表")]

public List<CLS\_GuildInfoBase> ListGuildInfoBase = new List<CLS\_GuildInfoBase>();

### 协议类 C2G\_Guild\_AllianceApply 请求 同盟申请

对应ID： C2G\_GUILD\_ALLIANCEAPPLY

[Desc("唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Guild\_AllianceApply 返回 同盟申请

对应ID： G2C\_GUILD\_ALLIANCEAPPLY

### 协议类 C2G\_Guild\_AllianceRemove 请求 同盟解除

对应ID： C2G\_GUILD\_ALLIANCEREMOVE

[Desc("唯一ID")]

public long Uid = 0;

### 协议类 G2C\_Guild\_AllianceRemove 返回 同盟解除

对应ID： G2C\_GUILD\_ALLIANCEREMOVE

### 协议类 C2G\_Guild\_AllianceApplyAction 请求 同盟申请处理

对应ID： C2G\_GUILD\_ALLIANCEAPPLYACTION

[Desc("对象ID")]

public long Uid = 0;

[Desc("处理方式 EHandleAction")]

public int HandleAction = 0;

### 协议类 G2C\_Guild\_AllianceApplyAction 返回 同盟申请处理

对应ID： G2C\_GUILD\_ALLIANCEAPPLYACTION

### 协议类 C2G\_Guild\_ListCity 请求 城池列表

对应ID： C2G\_GUILD\_LISTCITY

### 协议类 G2C\_Guild\_ListCity 返回 城池列表

对应ID： G2C\_GUILD\_LISTCITY

[Desc("都城ID")]

public int CapitalCity = 0;

[Desc("城池列表")]

public List<CLS\_CityInfo4Guild> ListCity = new List<CLS\_CityInfo4Guild>();

### 协议类 C2G\_Guild\_SetCapitalCity 请求 设置都城

对应ID： C2G\_GUILD\_SETCAPITALCITY

[Desc("城池ID")]

public int Uid = 0;

### 协议类 G2C\_Guild\_SetCapitalCity 返回 设置都城

对应ID： G2C\_GUILD\_SETCAPITALCITY

### 协议类 C2G\_Guild\_SetCityLeader 请求 任命太守

对应ID： C2G\_GUILD\_SETCITYLEADER

[Desc("城池ID")]

public int Uid = 0;

[Desc("新任命的玩家ID")]

public long LeaderPuid = 0;

### 协议类 G2C\_Guild\_SetCityLeader 返回 任命太守

对应ID： G2C\_GUILD\_SETCITYLEADER

### 协议类 C2G\_Guild\_RemoveCityLeader 请求 撤销太守

对应ID： C2G\_GUILD\_REMOVECITYLEADER

[Desc("城池ID")]

public int Uid = 0;

### 协议类 G2C\_Guild\_RemoveCityLeader 返回 撤销太守

对应ID： G2C\_GUILD\_REMOVECITYLEADER

### 协议类 C2G\_Guild\_ListLogSituation 请求 军情

对应ID： C2G\_GUILD\_LISTLOGSITUATION

### 协议类 G2C\_Guild\_ListLogSituation 返回 军情

对应ID： G2C\_GUILD\_LISTLOGSITUATION

[Desc("军情列表")]

public List<CLS\_GameLog> ListLog = new List<CLS\_GameLog>();

## 红包

### 公共类 CLS\_RedPacketMember 红包成员信息

[Desc("玩家ID")]

public long PlayerID = 0;

[Desc("头像ID")]

public int HeadId = 0;

public string Name = "";

[Desc("抢到金额")]

public int Bonus = 0;

### 公共类 CLS\_RedPacketsInfo //红包列表

[Desc("红包ID")]

public long RedPacketsID = 0;

[Desc("发起者")]

public CLS\_RedPacketMember Owner = new CLS\_RedPacketMember();

[Desc("状态 ERedPacketState")]

public int State = 0;

[Desc("还剩多少时间(秒)")]

public int TimeEnd = 0;

### 公共类 CLS\_RedPacket //红包信息

[Desc("红包ID")]

public long RedPacketsID = 0;

[Desc("总金额")]

public int TotalSugar = 0;

[Desc("总个数")]

public int TotalNumber = 0;

[Desc("剩余金额")]

public int SurplusSugar = 0;

[Desc("剩余个数")]

public int SurplusNumber = 0;

[Desc("发起者")]

public CLS\_RedPacketMember Owner = new CLS\_RedPacketMember();

[Desc("还剩多少时间(秒)")]

public int TimeEnd = 0;

[Desc("已抢列表")]

public List<CLS\_RedPacketMember> ListMember = new List<CLS\_RedPacketMember>();

[Desc("祝福语")]

public string Txt = "";

### 协议类 C2G\_Red\_Packet\_List 请求 红包列表

对应ID： C2G\_RED\_PACKET\_LIST

### 协议类 G2C\_Red\_Packet\_List 返回 红包列表

对应ID： G2C\_RED\_PACKET\_LIST

[Desc("0:领红包未达到上限,1:领红包达到上限")]

public int GetMax = 0;

[Desc("红包列表")]

public List<CLS\_RedPacketsInfo> Info = new List<CLS\_RedPacketsInfo>();

### 协议类 C2G\_Red\_Packet\_Info 请求 单个红包信息

对应ID： C2G\_RED\_PACKET\_INFO

[Desc("红包ID")]

public long RedPacketsID = 0;

### 协议类 G2C\_Red\_Packet\_Info 返回 单个红包信息

对应ID： G2C\_RED\_PACKET\_INFO

[Desc("状态")]

public int Error = 0;

[Desc("红包信息")]

public CLS\_RedPacket Info = new CLS\_RedPacket();

### 协议类 C2G\_Give\_Red\_Packet 请求 发红包

对应ID： C2G\_GIVE\_RED\_PACKET

[Desc("金额")]

public int Sugar = 0;

[Desc("个数")]

public int Number = 0;

[Desc("祝福语")]

public string Txt = "";

### 协议类 G2C\_Give\_Red\_Packet 返回 发红包

对应ID： G2C\_GIVE\_RED\_PACKET

### 协议类 G2C\_Red\_Packet\_Receive 返回 发红包

对应ID： G2C\_RED\_PACKET\_RECEIVE

[Desc("红包信息")]

public CLS\_RedPacketsInfo Info = new CLS\_RedPacketsInfo();

[Desc("红包列表")]

public List<CLS\_RedPacketsInfo> InfoList = new List<CLS\_RedPacketsInfo>();

### 协议类 C2G\_Get\_Red\_Packet 请求 抢红包

对应ID： C2G\_GET\_RED\_PACKET

[Desc("红包ID")]

public long RedPacketsID = 0;

### 协议类 G2C\_Get\_Red\_Packet 返回 抢红包

对应ID： G2C\_GET\_RED\_PACKET

[Desc("状态")]

public int Error = 0;

[Desc("抢到的金额")]

public int Bonus = 0;

[Desc("0:领红包未达到上限,1:领红包达到上限")]

public int GetMax = 0;

[Desc("红包信息")]

public CLS\_RedPacket Info = new CLS\_RedPacket();

## 府Boss

### 协议类 C2G\_MansionBoss\_Info 请求 获取府Boss信息

对应ID： C2G\_MANSIONBOSS\_INFO

[Desc("府ID")]

public long MansionId = 0;

### 协议类 G2C\_MansionBoss\_Info 返回 获取府Boss信息

对应ID： G2C\_MANSIONBOSS\_INFO

[Desc("是否已开启")]

public bool IsOpen = false;

[Desc("今日可开启次数")]

public int OpenCount = 0;

[Desc("ConfigID")]

public int ConfigID = 0;

[Desc("剩余毫秒")]

public long EndTs = 0;

[Desc("本日可挑战次数")]

public int Count = 0;

[Desc("最大可挑战次数")]

public int CountMax = 0;

[Desc("当前血量")]

public long HpCur = 0;

[Desc("玩家排名")]

public int Rank = 0;

### 协议类 C2G\_MansionBoss\_Open 请求 处理府Boss开启

对应ID： C2G\_MANSIONBOSS\_OPEN

[Desc("府ID")]

public long MansionId = 0;

### 协议类 G2C\_MansionBoss\_Open 返回 处理府Boss开启

对应ID： G2C\_MANSIONBOSS\_OPEN

### 公共类 CLS\_MansionBoss\_PlayerInfo Boss排行玩家信息

[Desc("玩家ID")]

public long Id = 0;

[Desc("头像")]

public int Headindex = 0;

[Desc("名字")]

public string Name = "";

[Desc("伤害")]

public long Hurt = 0;

[Desc("伤害百分比")]

public float HurtPer = 0.0f;

### 协议类 C2G\_MansionBoss\_Top 请求 获取府Boss伤害排行

对应ID： C2G\_MANSIONBOSS\_TOP

[Desc("府ID")]

public long MansionId = 0;

### 协议类 G2C\_MansionBoss\_Top 返回 获取府Boss伤害排行

对应ID： G2C\_MANSIONBOSS\_TOP

[Desc("玩家排行")]

public List<CLS\_MansionBoss\_PlayerInfo> TopPlayers = new List<CLS\_MansionBoss\_PlayerInfo>();

[Desc("玩家自己")]

public CLS\_MansionBoss\_PlayerInfo player = new CLS\_MansionBoss\_PlayerInfo();

[Desc("玩家排名")]

public int Rank = 0;

### 协议类 C2G\_MansionBoss\_Battle 请求 处理府Boss进入战斗

对应ID： C2G\_MANSIONBOSS\_BATTLE

[Desc("府ID")]

public long MansionId = 0;

### 协议类 G2C\_MansionBoss\_Battle 返回 处理府Boss进入战斗

对应ID： G2C\_MANSIONBOSS\_BATTLE

### 协议类 C2G\_MansionBoss\_Balance 请求 处理府Boss战斗结算

对应ID： C2G\_MANSIONBOSS\_BALANCE

[Desc("府ID")]

public long MansionId = 0;

[Desc("造成伤害")]

public long HurtHp = 0;

[Desc("战斗Key")]

public long BattleKey = 0;

[Desc("玩家消耗兵力")]

public int PlayerExpendHp = 0;

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHp = new Dictionary<long, int>();

[Desc("战斗结果")]

public int State = 0;

### 协议类 G2C\_MansionBoss\_Balance 返回 处理府Boss战斗结算

对应ID： G2C\_MANSIONBOSS\_BALANCE

### 协议类 G2C\_MansionBoss\_TakeBalance 返回 处理府Boss主动战斗结算

对应ID： G2C\_MANSIONBOSS\_TAKEBALANCE

### 协议类 C2G\_MansionBoss\_Buff 请求 处理府BossBuff信息和购买

对应ID： C2G\_MANSIONBOSS\_BUFF

[Desc("府ID")]

public long MansionId = 0;

[Desc("购买buff配置ID 为0 则是查看信息")]

public int Config = 0;

### 协议类 G2C\_MansionBoss\_Buff 返回 处理府BossBuff信息和购买

对应ID： G2C\_MANSIONBOSS\_BUFF

[Desc("Buffid")]

public List<int> BuffIds = new List<int>();

[Desc("已经拥有的buffid，剩余时间")]

public Dictionary<int, long> CurBuff = new Dictionary<int, long>();

[Desc("拥有的府玉")]

public long FuYu = 0;

## 势力活跃

### 公共类 CLS\_Liveness\_Info 单条活跃状态

[Desc("数值")]

public int Num = 0;

[Desc("是否已经领取")]

public Dictionary<int, bool> DictAwardState = new Dictionary<int, bool>();

### 协议类 C2G\_Guild\_GetActive 请求 府活跃度

对应ID： C2G\_GUILD\_GETACTIVE

[Desc("府id")]

public long GuildId = 0;

### 协议类 G2C\_Guild\_GetActive 返回 府活跃度

对应ID： G2C\_GUILD\_GETACTIVE

[Desc("活跃状态")]

public Dictionary<int, CLS\_Liveness\_Info> DictLivenessInfo = new Dictionary<int, CLS\_Liveness\_Info>();

### 协议类 C2G\_Guild\_ActiveAward 请求 府活跃度奖励领取

对应ID： C2G\_GUILD\_ACTIVEAWARD

[Desc("府id")]

public long GuildId = 0;

[Desc("箱子编号")]

public int BoxID = 0;

### 协议类 G2C\_Guild\_ActiveAward 返回 府活跃度奖励领取

对应ID： G2C\_GUILD\_ACTIVEAWARD

[Desc("奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

[Desc("活跃状态")]

public Dictionary<int, CLS\_Liveness\_Info> DictLivenessInfo = new Dictionary<int, CLS\_Liveness\_Info>();

### 协议类 C2G\_Guild\_ActiveAwardAll 请求 府活跃度奖励全部领取

对应ID： C2G\_GUILD\_ACTIVEAWARDALL

### 协议类 G2C\_Guild\_ActiveAwardAll 返回 府活跃度奖励全部领取

对应ID： G2C\_GUILD\_ACTIVEAWARDALL

[Desc("奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

[Desc("活跃状态")]

public Dictionary<int, CLS\_Liveness\_Info> DictLivenessInfo = new Dictionary<int, CLS\_Liveness\_Info>();

## 势力科技和商店

### 协议类 C2G\_Guild\_ScienceInfo 请求 书院信息

对应ID： C2G\_GUILD\_SCIENCEINFO

### 协议类 G2C\_Guild\_ScienceInfo 返回 书院信息

对应ID： G2C\_GUILD\_SCIENCEINFO

[Desc("科技ID列表")]

public List<int> ListTechnology = new List<int>();

### 协议类 C2G\_Guild\_ScienceLevelUp 请求 势力科技升级

对应ID： C2G\_GUILD\_SCIENCELEVELUP

[Desc("科技ID")]

public int ID = 0;

### 协议类 G2C\_Guild\_ScienceLevelUp 返回 势力科技升级

对应ID： G2C\_GUILD\_SCIENCELEVELUP

### 公共类 CLS\_GuildShopInfo 势力商店信息

[Desc("商品信息")]

public List<CLS\_ShopGoodsInfo> GoodsInfo = new List<CLS\_ShopGoodsInfo>();

[Desc("拥有的贡献")]

public int Contribution = 0;

[Desc("累计免费刷新次数")]

public int RefreshFreeTotal = 0;

[Desc("累计刷新次数")]

public int RefreshTotal = 0;

[Desc("刷新道具数量")]

public int RefreshItemCount = 0;

### 协议类 C2G\_Guild\_ShopInfo 请求 处理势力商店信息

对应ID： C2G\_GUILD\_SHOPINFO

### 协议类 G2C\_Guild\_ShopInfo 返回 处理势力商店信息

对应ID： G2C\_GUILD\_SHOPINFO

[Desc("商品信息")]

public CLS\_GuildShopInfo ShopInfo = new CLS\_GuildShopInfo();

### 协议类 C2G\_Guild\_ShopBuy 请求 处理势力商店购买

对应ID： C2G\_GUILD\_SHOPBUY

[Desc("配置ID")]

public int Config = 0;

[Desc("数量")]

public int Amount = 0;

### 协议类 G2C\_Guild\_ShopBuy 返回 处理势力商店购买

对应ID： G2C\_GUILD\_SHOPBUY

[Desc("商品信息")]

public CLS\_GuildShopInfo ShopInfo = new CLS\_GuildShopInfo();

### 协议类 C2G\_Guild\_ShopRefresh 请求 处理势力商店刷新

对应ID： C2G\_GUILD\_SHOPREFRESH

### 协议类 G2C\_Guild\_ShopRefresh 返回 处理势力商店刷新

对应ID： G2C\_GUILD\_SHOPREFRESH

[Desc("商品信息")]

public CLS\_GuildShopInfo ShopInfo = new CLS\_GuildShopInfo();

## 势力俸禄

### 公共类 CLS\_GuildSalaryMbsInfo 势力成员俸禄信息

[Desc("基本信息")]

public CLS\_PubPlayerBase PubPlayerBase = new CLS\_PubPlayerBase();

[Desc("上周贡献")]

public int ContributionWeekLast = 0;

[Desc("本周贡献")]

public int ContributionWeekThis = 0;

### 协议类 C2G\_GuildSalary\_ListMbs 请求 俸禄成员列表

对应ID： C2G\_GUILDSALARY\_LISTMBS

### 协议类 G2C\_GuildSalary\_ListMbs 返回 俸禄成员列表

对应ID： G2C\_GUILDSALARY\_LISTMBS

[Desc("自动结算时间")]

public DateTime DtAutoBalance = DateTime.Now;

[Desc("势力成员列表")]

public List<CLS\_GuildSalaryMbsInfo> ListMember = new List<CLS\_GuildSalaryMbsInfo>();

### 协议类 C2G\_GuildSalary\_Pay 请求 俸禄发放

对应ID： C2G\_GUILDSALARY\_PAY

[Desc("<对象玩家ID, 金币数量>")]

public Dictionary<long, int> DictPay = new Dictionary<long, int>();

### 协议类 G2C\_GuildSalary\_Pay 返回 俸禄发放

对应ID： G2C\_GUILDSALARY\_PAY

[Desc("<对象玩家ID, 金币数量>")]

public Dictionary<long, int> DictPay = new Dictionary<long, int>();

### 公共类 CLS\_GuildSalaryCityInfo 势力城市俸禄信息

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("太守Id")]

public long LeaderPuid = 0;

[Desc("太守名字")]

public string LeaderName = "";

[Desc("繁荣度")]

public long Prosperity = 0;

[Desc("预期收益")]

public int Revenue = 0;

### 协议类 C2G\_GuildSalary\_ListCity 请求 俸禄城市列表

对应ID： C2G\_GUILDSALARY\_LISTCITY

### 协议类 G2C\_GuildSalary\_ListCity 返回 俸禄城市列表

对应ID： G2C\_GUILDSALARY\_LISTCITY

[Desc("势力城市列表")]

public List<CLS\_GuildSalaryCityInfo> ListCity = new List<CLS\_GuildSalaryCityInfo>();

## PVE

### 协议类 C2G\_Warrior\_InBattle 请求 武将上阵PVE

对应ID： C2G\_WARRIOR\_INBATTLE

[Desc("上阵列表")]

public List<long> ListInBattle = new List<long>();

### 协议类 G2C\_Warrior\_InBattle 返回 武将上阵PVE

对应ID： G2C\_WARRIOR\_INBATTLE

### 公共类 CLS\_ArmyInfo 单个队伍信息

[Desc("PVE阵容ID(012)")]

public int ArmyId = 0;

[Desc("成员列表 <位置 武将唯一ID>")]

public Dictionary<int, long> DictWarrior = new Dictionary<int, long>();

### 公共类 CLS\_PveTypeArmyInfo PVE某种战斗类型队伍信息

[Desc("PVE类型(ETeamType)")]

public int PveType = 0;

[Desc("当前PVE阵容ID(012)")]

public int PveArmyId = 0;

[Desc("k=ArmyId v=ArmyInfo")]

public Dictionary<int, CLS\_ArmyInfo> DictArmyInfo = new Dictionary<int, CLS\_ArmyInfo>();

### 协议类 C2G\_Pve\_ArmyInfo 请求 PVE阵容信息

对应ID： C2G\_PVE\_ARMYINFO

[Desc("PVE类型")]

public int PveType = 0;

### 协议类 G2C\_Pve\_ArmyInfo 返回 PVE阵容信息

对应ID： G2C\_PVE\_ARMYINFO

[Desc("PVE类型(ETeamType)")]

public int PveType = 0;

[Desc("当前PVE阵容ID(012)")]

public int PveArmyId = 0;

[Desc("k=ArmyId v=ArmyInfo")]

public Dictionary<int, CLS\_ArmyInfo> DictArmyInfo = new Dictionary<int, CLS\_ArmyInfo>();

### 协议类 C2G\_Pve\_ArmyChange 请求 PVE阵容更改

对应ID： C2G\_PVE\_ARMYCHANGE

[Desc("PVE类型(ETeamType)")]

public int PveType = 0;

[Desc("当前PVE阵容ID(012)")]

public int PveArmyId = 0;

[Desc("k=ArmyId v=ArmyInfo")]

public Dictionary<int, CLS\_ArmyInfo> DictArmyInfo = new Dictionary<int, CLS\_ArmyInfo>();

### 协议类 G2C\_Pve\_ArmyChange 返回 PVE阵容更改

对应ID： G2C\_PVE\_ARMYCHANGE

[Desc("PVE类型(ETeamType)")]

public int PveType = 0;

### 协议类 C2G\_Pve\_GoBattle 请求 PVE出征

对应ID： C2G\_PVE\_GOBATTLE

[Desc("PVE类型(ETeamType)")]

public int PveType = 0;

[Desc("对应战斗内容ID")]

public int BattleId = 0;

### 协议类 G2C\_Pve\_GoBattle 返回 PVE出征

对应ID： G2C\_PVE\_GOBATTLE

[Desc("PVE类型(ETeamType)")]

public int PveType = 0;

[Desc("对应战斗内容ID")]

public int BattleId = 0;

[Desc("战斗验证码")]

public long BattleCode = 0;

[Desc("成员列表 <位置 武将战斗信息>")]

public Dictionary<int, CLS\_WarriorInfo> DictWarrior = new Dictionary<int, CLS\_WarriorInfo>();

### 协议类 G2C\_Pve\_BattleBalance 推送 战斗结束武将数据

对应ID： G2C\_PVE\_BATTLEBALANCE

[Desc("PVE类型(ETeamType)")]

public int PveType = 0;

[Desc("成员列表 <位置 武将战斗信息>")]

public Dictionary<int, CLS\_WarriorInfo> DictWarrior = new Dictionary<int, CLS\_WarriorInfo>();

## 关卡

### 协议类 C2G\_GameLevelDungeon\_Info 请求 校场拉练关卡信息

对应ID： C2G\_GAMELEVELDUNGEON\_INFO

### 协议类 G2C\_GameLevelDungeon\_Info 返回 校场拉练关卡信息

对应ID： G2C\_GAMELEVELDUNGEON\_INFO

[Desc("当前战斗关卡")]

public int Id = 0;

### 协议类 C2G\_GameLevelDungeon\_Battle 请求 校场拉练关卡战斗

对应ID： C2G\_GAMELEVELDUNGEON\_BATTLE

### 协议类 G2C\_GameLevelDungeon\_Battle 返回 校场拉练关卡战斗

对应ID： G2C\_GAMELEVELDUNGEON\_BATTLE

[Desc("关卡ID")]

public int Id = 0;

### 协议类 C2G\_GameLevelDungeon\_Balance 请求 校场拉练关卡结算

对应ID： C2G\_GAMELEVELDUNGEON\_BALANCE

[Desc("战斗Key")]

public long BattleKey = 0;

[Desc("玩家消耗兵力")]

public int PlayerExpendHp = 0;

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHp = new Dictionary<long, int>();

[Desc("战斗结果")]

public int State = 0;

### 协议类 G2C\_GameLevelDungeon\_Balance 返回 校场拉练关卡结算

对应ID： G2C\_GAMELEVELDUNGEON\_BALANCE

[Desc("结算奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 公共类 CLS\_StoryState 剧情关卡结果

[Desc("战斗结果")]

public List<bool> ListState = new List<bool>();

[Desc("日挑战次数")]

public int Ticks = 0;

### 协议类 C2G\_GameLevelStory\_Info 请求 剧情关卡列表

对应ID： C2G\_GAMELEVELSTORY\_INFO

[Desc("章节ID")]

public int StoryID = 0;

### 协议类 G2C\_GameLevelStory\_Info 返回 剧情关卡列表

对应ID： G2C\_GAMELEVELSTORY\_INFO

[Desc("副本字典<关卡ID，通关状态>")]

public Dictionary<int, CLS\_StoryState> DicStory = new Dictionary<int, CLS\_StoryState>();

[Desc("体力")]

public int Power = 0;

[Desc("恢复时间")]

public long RecoveryTime = 0;

[Desc("奖励是否已经领取")]

public List<bool> ListReward = new List<bool>();

[Desc("购买次数")]

public int BuyCount = 0;

### 协议类 C2G\_GameLevelStory\_Battle 请求 剧情副本战斗

对应ID： C2G\_GAMELEVELSTORY\_BATTLE

[Desc("副本ID")]

public int Id = 0;

### 协议类 G2C\_GameLevelStory\_Battle 返回 剧情副本战斗

对应ID： G2C\_GAMELEVELSTORY\_BATTLE

[Desc("关卡ID")]

public int Id = 0;

### 协议类 C2G\_GameLevelStory\_Balance 请求 剧情副本结算

对应ID： C2G\_GAMELEVELSTORY\_BALANCE

[Desc("战斗Key")]

public long BattleKey = 0;

[Desc("玩家消耗兵力")]

public int PlayerExpendHp = 0;

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHp = new Dictionary<long, int>();

[Desc("战斗星级结果")]

public List<bool> ListState = new List<bool>();

[Desc("战斗结果")]

public int State = 0;

### 协议类 G2C\_GameLevelStory\_Balance 返回 剧情副本结算

对应ID： G2C\_GAMELEVELSTORY\_BALANCE

[Desc("结算奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_GameLevelStory\_Reward 请求 剧情副本领取奖励

对应ID： C2G\_GAMELEVELSTORY\_REWARD

[Desc("章节ID")]

public int StoryID = 0;

[Desc("箱子编号（0,1,2）")]

public int Index = 0;

### 协议类 G2C\_GameLevelStory\_Reward 返回 剧情副本领取奖励

对应ID： G2C\_GAMELEVELSTORY\_REWARD

[Desc("结算奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 C2G\_GameLevelStoryBuyPower 请求 剧情副本购买体力

对应ID： C2G\_GAMELEVELSTORYBUYPOWER

### 协议类 G2C\_GameLevelStoryBuyPower 返回 剧情副本购买体力

对应ID： G2C\_GAMELEVELSTORYBUYPOWER

[Desc("体力")]

public int Power = 0;

[Desc("恢复时间")]

public long RecoveryTime = 0;

[Desc("购买次数")]

public int BuyCount = 0;

### 协议类 C2G\_GameLevelStory 请求 剧情章节列表

对应ID： C2G\_GAMELEVELSTORY

### 协议类 G2C\_GameLevelStory 返回 剧情章节列表

对应ID： G2C\_GAMELEVELSTORY

[Desc("当前章节ID")]

public int StoryID = 0;

[Desc("体力")]

public int Power = 0;

[Desc("恢复时间")]

public long RecoveryTime = 0;

[Desc("购买次数")]

public int BuyCount = 0;

## 古塔

### 公共类 CLS\_TowerInfo 单层信息

[Desc("ConfigId")]

public int Id = 0;

[Desc("ETowerState")]

public byte TowerState = 0;

### 协议类 C2G\_Tower\_Info 请求 古塔信息

对应ID： C2G\_TOWER\_INFO

### 协议类 G2C\_Tower\_Info 返回 古塔信息

对应ID： G2C\_TOWER\_INFO

[Desc("可进入等级")]

public int LevelStart = 0;

[Desc("挑战令牌")]

public int TowerToken = 0;

[Desc("挑战令牌上限")]

public int TowerTokenMax = 0;

[Desc("古塔挑战令牌单次消耗")]

public int TowerTokenOnceUse = 0;

[Desc("层信息")]

public List<CLS\_TowerInfo> ListTowerInfo = new List<CLS\_TowerInfo>();

### 协议类 C2G\_Tower\_Battle 请求 古塔战斗

对应ID： C2G\_TOWER\_BATTLE

[Desc("ConfigId")]

public int Id = 0;

### 协议类 G2C\_Tower\_Battle 返回 古塔战斗

对应ID： G2C\_TOWER\_BATTLE

[Desc("ConfigId")]

public int Id = 0;

### 协议类 C2G\_Tower\_BattleEnd 请求 古塔战斗结束

对应ID： C2G\_TOWER\_BATTLEEND

[Desc("ConfigId")]

public int Id = 0;

[Desc("战斗Key")]

public long BattleKey = 0;

[Desc("玩家消耗兵力")]

public int PlayerExpendHp = 0;

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHp = new Dictionary<long, int>();

[Desc("战斗结果")]

public int State = 0;

[Desc("战斗记录")]

public string BattleRecord = "";

### 协议类 G2C\_Tower\_BattleEnd 返回 古塔战斗结束

对应ID： G2C\_TOWER\_BATTLEEND

[Desc("ConfigId")]

public int Id = 0;

[Desc("奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

## Arena

### 公共类 CLS\_ArenaTopData Arena排行榜信息

[Desc("排行")]

public int Rank = 0;

[Desc("唯一ID")]

public long Puid = 0;

[Desc("名字")]

public string Name = "";

[Desc("势力名字")]

public string GuildName = "";

[Desc("分数")]

public int Score = 0;

[Desc("头像ID")]

public int HeadIndex = 0;

[Desc("等级 排行榜用")]

public int Level = 0;

### 协议类 C2G\_Arena\_Info 请求 获取竞技场信息

对应ID： C2G\_ARENA\_INFO

### 协议类 G2C\_Arena\_Info 返回 获取竞技场信息

对应ID： G2C\_ARENA\_INFO

[Desc("排名")]

public int Rank = 0;

[Desc("历史排名")]

public int HistoryRank = 0;

[Desc("胜率")]

public float WinRate = 0.0f;

[Desc("挑战次数")]

public int MatchTicks = 0;

[Desc("最大挑战次数")]

public int MaxTicks = 0;

[Desc("竞技场已购买次数")]

public int BuyMatchTicks = 0;

[Desc("匹配的玩家")]

public List<CLS\_ArenaTopData> ArenaMatchPlayers = new List<CLS\_ArenaTopData>();

[Desc("挑战刷新剩余时间")]

public long TsEnd = 0;

### 协议类 C2G\_Arena\_TOP 请求 获取竞技场排行榜

对应ID： C2G\_ARENA\_TOP

### 协议类 G2C\_Arena\_TOP 返回 Arena排行榜

对应ID： G2C\_ARENA\_TOP

[Desc("Arena排行榜数据")]

public List<CLS\_ArenaTopData> ListArenaTopData = new List<CLS\_ArenaTopData>();

[Desc("玩家数据")]

public CLS\_ArenaTopData PlayerData = new CLS\_ArenaTopData();

### 协议类 C2G\_ArenaMatch\_Refresh 请求 Arena刷新

对应ID： C2G\_ARENAMATCH\_REFRESH

### 协议类 G2C\_ArenaMatch\_Refresh 返回 Arena刷新

对应ID： G2C\_ARENAMATCH\_REFRESH

### 协议类 C2G\_ArenaMatch\_Battle 请求 Arena战斗

对应ID： C2G\_ARENAMATCH\_BATTLE

[Desc("挑战的第几位玩家 0,1,2,3")]

public int index = 0;

### 协议类 G2C\_ArenaMatch\_Battle 返回 Arena战斗

对应ID： G2C\_ARENAMATCH\_BATTLE

[Desc("关卡配置 0则是玩家")]

public int StageId = 0;

[Desc("对战武将战斗信息列表")]

public List<CLS\_WarriorInfo> ListWarriorOther = new List<CLS\_WarriorInfo>();

### 协议类 C2G\_ArenaMatch\_Balance 请求 Arena结算

对应ID： C2G\_ARENAMATCH\_BALANCE

[Desc("战斗Key")]

public long BattleKey = 0;

[Desc("战斗结果")]

public int State = 0;

### 协议类 G2C\_ArenaMatch\_Balance 返回 Arena结算

对应ID： G2C\_ARENAMATCH\_BALANCE

[Desc("玩家当前排名")]

public int PlayerRank = 0;

[Desc("玩家新排名")]

public int PlayerNewRank = 0;

[Desc("奖励")]

public List<CLS\_AwardItem> AwardItem = new List<CLS\_AwardItem>();

### 公共类 CLS\_ArenaBattleReport Arena战报单条信息

[Desc("胜利")]

public bool Victory = false;

[Desc("是否是进攻")]

public bool IsAttack = false;

[Desc("名字")]

public string MatchPlayerName = "";

[Desc("头像")]

public int MatchPlayerHeadIndex = 0;

[Desc("势力名称")]

public string GuildName = "";

[Desc("排名变化")]

public float Rank = 0.0f;

[Desc("当前排名")]

public int CurRank = 0;

### 协议类 C2G\_ArenaBattleReport 请求 Arena战报

对应ID： C2G\_ARENABATTLEREPORT

### 协议类 G2C\_ArenaBattleReport 返回 Arena战报

对应ID： G2C\_ARENABATTLEREPORT

[Desc("战报列表")]

public List<CLS\_ArenaBattleReport> ArenaBattleReportList = new List<CLS\_ArenaBattleReport>();

### 协议类 C2G\_ArenaMatch\_Buy 请求 竞技场次数购买

对应ID： C2G\_ARENAMATCH\_BUY

### 协议类 G2C\_ArenaMatch\_Buy 返回 竞技场次数购买

对应ID： G2C\_ARENAMATCH\_BUY

[Desc("挑战次数")]

public int MatchTicks = 0;

### 协议类 C2G\_Arena\_Defense 请求 查看竞技场防守队伍

对应ID： C2G\_ARENA\_DEFENSE

### 协议类 G2C\_Arena\_Defense 返回 查看竞技场防守队伍

对应ID： G2C\_ARENA\_DEFENSE

[Desc("防守队伍信息<位置，信息>")]

public Dictionary<int, CLS\_WarriorInfo> DictWarriorInfo = new Dictionary<int, CLS\_WarriorInfo>();

### 协议类 C2G\_Arena\_SetDefense 请求 竞技场设置防守队伍

对应ID： C2G\_ARENA\_SETDEFENSE

[Desc("防守队伍<位置，ID>")]

public Dictionary<int, long> DictDefenses = new Dictionary<int, long>();

### 协议类 G2C\_Arena\_SetDefense 返回 竞技场设置防守队伍

对应ID： G2C\_ARENA\_SETDEFENSE

## 长安夺宝

### 公共类 CLS\_BankInfo 长安夺宝信息

[Desc("是否已到活动时间")]

public bool IsCanFight = false;

[Desc("是否已经报名")]

public bool IsSignUp = false;

[Desc("金币")]

public long GoldCoin = 0;

[Desc("每日挑战次数")]

public int ActiveTick = 0;

[Desc("总共挑战次数")]

public int ActiveAllTick = 0;

### 协议类 C2G\_Bank\_Info 请求 长安夺宝面板

对应ID： C2G\_BANK\_INFO

### 协议类 G2C\_Bank\_Info 返回 长安夺宝面板

对应ID： G2C\_BANK\_INFO

[Desc("长安夺宝信息")]

public CLS\_BankInfo BankInfo = new CLS\_BankInfo();

### 协议类 C2G\_Bank\_SignIn 请求 长安夺宝报名

对应ID： C2G\_BANK\_SIGNIN

[Desc("战区")]

public int WarZone = 0;

### 协议类 G2C\_Bank\_SignIn 返回 长安夺宝报名

对应ID： G2C\_BANK\_SIGNIN

[Desc("长安夺宝信息")]

public CLS\_BankInfo BankInfo = new CLS\_BankInfo();

### 公共类 CLS\_WarZoneInfo 战区关卡信息

[Desc("配置ID")]

public int ConfigID = 0;

[Desc("守军关卡状态")]

public EBankDefend BankDefend = new EBankDefend();

[Desc("剩余血量")]

public long Hp = 0;

### 协议类 C2G\_Bank\_WarZone 请求 长安夺宝战区信息

对应ID： C2G\_BANK\_WARZONE

### 协议类 G2C\_Bank\_WarZone 返回 长安夺宝战区信息

对应ID： G2C\_BANK\_WARZONE

[Desc("战区信息")]

public List<CLS\_WarZoneInfo> ListWarZone = new List<CLS\_WarZoneInfo>();

[Desc("贡献值")]

public List<CLS\_BankCountryTop> ListContribution = new List<CLS\_BankCountryTop>();

[Desc("自国家")]

public int CountryId = 0;

[Desc("贡献奖励领取状态")]

public List<bool> ListWelfareState = new List<bool>();

[Desc("次数恢复时间点")]

public long RecoveryTime = 0;

[Desc("挑战次数")]

public long Ticks = 0;

### 协议类 C2G\_Bank\_WarZoneAward 请求 长安夺宝战区领奖

对应ID： C2G\_BANK\_WARZONEAWARD

[Desc("领奖ID")]

public int ConfigID = 0;

### 协议类 G2C\_Bank\_WarZoneAward 返回 长安夺宝战区领奖

对应ID： G2C\_BANK\_WARZONEAWARD

[Desc("结算奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

[Desc("贡献奖励领取状态")]

public List<bool> ListWelfareState = new List<bool>();

### 协议类 C2G\_Bank\_WarZoneFight 请求 长安夺宝战区战斗

对应ID： C2G\_BANK\_WARZONEFIGHT

[Desc("据点ID")]

public int StrongHold = 0;

### 协议类 G2C\_Bank\_WarZoneFight 返回 长安夺宝战区战斗

对应ID： G2C\_BANK\_WARZONEFIGHT

### 协议类 C2G\_Bank\_WarZoneBalance 请求 长安夺宝战区战斗结算

对应ID： C2G\_BANK\_WARZONEBALANCE

[Desc("造成伤害")]

public long HurtHp = 0;

[Desc("战斗Key")]

public long BattleKey = 0;

### 协议类 G2C\_Bank\_WarZoneBalance 返回 长安夺宝战区战斗结算

对应ID： G2C\_BANK\_WARZONEBALANCE

### 协议类 C2G\_Bank\_WarZoneHurtHp 请求 长安夺宝战区战斗伤害(每秒计算)

对应ID： C2G\_BANK\_WARZONEHURTHP

[Desc("造成伤害")]

public long HurtHp = 0;

[Desc("战斗Key")]

public long BattleKey = 0;

### 协议类 G2C\_Bank\_WarZoneHurtHp 返回 长安夺宝战区战斗伤害

对应ID： G2C\_BANK\_WARZONEHURTHP

### 协议类 C2G\_Bank\_RobberFight 请求 长安夺宝清缴匪盗

对应ID： C2G\_BANK\_ROBBERFIGHT

### 协议类 G2C\_Bank\_RobberFight 返回 长安夺宝清缴匪盗

对应ID： G2C\_BANK\_ROBBERFIGHT

### 协议类 C2G\_Bank\_RobberBalance 请求 长安夺宝清缴匪盗结算

对应ID： C2G\_BANK\_ROBBERBALANCE

[Desc("战斗Key")]

public long BattleKey = 0;

[Desc("战斗结果")]

public int State = 0;

### 协议类 G2C\_Bank\_RobberBalance 返回 长安夺宝清缴匪盗结算

对应ID： G2C\_BANK\_ROBBERBALANCE

[Desc("结算奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 公共类 CLS\_BankTop 长安夺宝排行

[Desc("唯一ID（玩家或者势力）")]

public long Uid = 0;

[Desc("伤害")]

public long HurtHp = 0;

[Desc("等级")]

public int Level = 0;

[Desc("玩家名")]

public string PlayerName = "";

[Desc("势力名")]

public string GuildName = "";

[Desc("国家")]

public int Country = 0;

[Desc("战区")]

public int WarZone = 0;

### 协议类 C2G\_Bank\_Top 请求 长安夺宝排行榜

对应ID： C2G\_BANK\_TOP

[Desc("战区ID")]

public int WarZoneID = 0;

### 协议类 G2C\_Bank\_Top 返回 长安夺宝排行榜

对应ID： G2C\_BANK\_TOP

[Desc("战区势力排行数据")]

public List<CLS\_BankTop> ListGuildTop = new List<CLS\_BankTop>();

[Desc("战区个人排行数据")]

public List<CLS\_BankTop> ListPlayerTop = new List<CLS\_BankTop>();

### 公共类 CLS\_BankCountryTop 长安夺宝国家排行

[Desc("国家ID")]

public int Countryid = 0;

[Desc("伤害")]

public long HurtHp = 0;

[Desc("国家前三名数据")]

public List<CLS\_PlayerData> TopThree = new List<CLS\_PlayerData>();

[Desc("国家个人排行数据")]

public List<CLS\_BankTop> ListPlayerTop = new List<CLS\_BankTop>();

### 协议类 C2G\_Bank\_TopCountry 请求 长安夺宝国家排行榜

对应ID： C2G\_BANK\_TOPCOUNTRY

### 协议类 G2C\_Bank\_TopCountry 返回 长安夺宝国家排行榜

对应ID： G2C\_BANK\_TOPCOUNTRY

[Desc("国家排行数据")]

public List<CLS\_BankCountryTop> ListTop = new List<CLS\_BankCountryTop>();

### 公共类 CLS\_BankConf 长安夺宝配置

[Desc("是否已设置")]

public bool IsSeted = false;

[Desc("金币")]

public long Gold = 0;

[Desc("玩家分配比例（50代表50%）")]

public int PlayerRatio = 0;

[Desc("报名时间")]

public DateTime TimeSignIn = DateTime.Now;

[Desc("开始时间")]

public DateTime TimeBegin = DateTime.Now;

[Desc("结束时间")]

public DateTime TimeEnd = DateTime.Now;

[Desc("每滴血的金币")]

public float HpGold = 0.0f;

## 军务

### 公共类 CLS\_Affairs 军务信息

[Desc("0-5普通 6每周")]

public int AffairsIndex = 0;

[Desc("配置ID")]

public int configid = 0;

[Desc("状态:EAffairsState")]

public int State = 0;

[Desc("开始时间")]

public DateTime StartTime = DateTime.Now;

[Desc("结束时间")]

public DateTime EndTime = DateTime.Now;

[Desc("武将职业")]

public List<int> ListJob = new List<int>();

[Desc("武将")]

public List<long> ListWarrior = new List<long>();

### 协议类 C2G\_Affairs\_List 请求 获取军务列表

对应ID： C2G\_AFFAIRS\_LIST

### 协议类 G2C\_Affairs\_List 返回 获取军务列表

对应ID： G2C\_AFFAIRS\_LIST

[Desc("军务列表")]

public List<CLS\_Affairs> ListAffairs = new List<CLS\_Affairs>();

### 协议类 C2G\_Affairs\_Begin 请求 开始军务

对应ID： C2G\_AFFAIRS\_BEGIN

[Desc("0-5普通 6每周")]

public int AffairsIndex = 0;

[Desc("武将")]

public List<long> ListWarrior = new List<long>();

### 协议类 G2C\_Affairs\_Begin 返回 开始军务

对应ID： G2C\_AFFAIRS\_BEGIN

[Desc("军务信息")]

public CLS\_Affairs AffairsInfo = new CLS\_Affairs();

### 协议类 C2G\_Affairs\_Award 请求 军务领奖

对应ID： C2G\_AFFAIRS\_AWARD

[Desc("0-5普通 6每周")]

public int AffairsIndex = 0;

### 协议类 G2C\_Affairs\_Award 返回 军务领奖

对应ID： G2C\_AFFAIRS\_AWARD

[Desc("奖励列表")]

public List<CLS\_AwardItem> ListAward = new List<CLS\_AwardItem>();

### 协议类 G2C\_Affairs\_Notify 通知军务完成

对应ID： G2C\_AFFAIRS\_NOTIFY

[Desc("刷新军务信息")]

public CLS\_Affairs AffairsInfo = new CLS\_Affairs();

## 资源点

### 公共类 CLS\_Grab\_ArmyInfo 矿点信息

[Desc("矿点ID")]

public int Id = 0;

[Desc("占领者")]

public long PalyerId = 0;

[Desc("占领者")]

public string Name = "";

[Desc("占领者所属势力")]

public string GuildName = "";

[Desc("队伍列表 <位置 武将唯一ID>")]

public Dictionary<int, long> DictWarrior = new Dictionary<int, long>();

[Desc("占领时间")]

public DateTime BeginTime = DateTime.Now;

[Desc("结束时间")]

public DateTime EndTime = DateTime.Now;

### 协议类 C2G\_Grab\_List 请求 资源点信息

对应ID： C2G\_GRAB\_LIST

[Desc("州ID")]

public int StateId = 0;

[Desc("资源类型")]

public int Type = 0;

### 协议类 G2C\_Grab\_List 返回 资源点信息

对应ID： G2C\_GRAB\_LIST

[Desc("购买次数")]

public int Buy = 0;

[Desc("剩余挑战次数")]

public int Total = 0;

public List<CLS\_Grab\_ArmyInfo> ListItem = new List<CLS\_Grab\_ArmyInfo>();

### 协议类 C2G\_Grab\_Battle 请求 矿点战斗

对应ID： C2G\_GRAB\_BATTLE

[Desc("矿点ID")]

public int Id = 0;

### 协议类 G2C\_Grab\_Battle 返回 矿点战斗

对应ID： G2C\_GRAB\_BATTLE

[Desc("对战武将战斗信息列表")]

public List<CLS\_WarriorInfo> ListWarriorOther = new List<CLS\_WarriorInfo>();

### 协议类 C2G\_Grab\_Fight 请求 占领矿点

对应ID： C2G\_GRAB\_FIGHT

[Desc("战斗ID")]

public long BattleUid = 0;

[Desc("战斗结果 攻击方胜利=true 防守方胜利=false")]

public bool BattleResult = false;

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHpAtt = new Dictionary<long, int>();

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHpDef = new Dictionary<long, int>();

### 协议类 G2C\_Grab\_Fight 返回 占领矿点

对应ID： G2C\_GRAB\_FIGHT

[Desc("胜利与否")]

public bool bSuccess = false;

### 协议类 C2G\_Grab\_Info 请求 查看矿点

对应ID： C2G\_GRAB\_INFO

[Desc("矿点ID")]

public int Id = 0;

### 协议类 G2C\_Grab\_Info 返回 查看矿点

对应ID： G2C\_GRAB\_INFO

[Desc("矿点ID")]

public int Id = 0;

[Desc("占领者")]

public long PalyerId = 0;

[Desc("占领者")]

public string Name = "";

[Desc("占领者所属势力")]

public string GuildName = "";

[Desc("防守队伍信息<位置，信息>")]

public Dictionary<int, CLS\_WarriorInfo> DictWarrior = new Dictionary<int, CLS\_WarriorInfo>();

[Desc("结束时间")]

public DateTime EndTime = DateTime.Now;

### 协议类 C2G\_Grab\_GiveUp 请求 放弃矿点

对应ID： C2G\_GRAB\_GIVEUP

[Desc("矿点ID")]

public int Id = 0;

### 协议类 G2C\_Grab\_GiveUp 返回 放弃矿点

对应ID： G2C\_GRAB\_GIVEUP

[Desc("矿点ID")]

public int Id = 0;

### 协议类 C2G\_Grab\_Mine 请求 自己的资源点信息

对应ID： C2G\_GRAB\_MINE

### 协议类 G2C\_Grab\_Mine 返回 自己的资源点信息

对应ID： G2C\_GRAB\_MINE

public List<CLS\_Grab\_ArmyInfo> ListItem = new List<CLS\_Grab\_ArmyInfo>();

### 协议类 C2G\_Grab\_Buy 请求 购买挑战次数

对应ID： C2G\_GRAB\_BUY

### 协议类 G2C\_Grab\_Buy 返回 购买挑战次数

对应ID： G2C\_GRAB\_BUY

[Desc("购买次数")]

public int Buy = 0;

[Desc("剩余挑战次数")]

public int Total = 0;

## 补兵

### 协议类 C2G\_Supply\_Supply 请求 补兵

对应ID： C2G\_SUPPLY\_SUPPLY

[Desc("k=武将唯一ID v=当前兵数")]

public Dictionary<long, int> DictWarrior = new Dictionary<long, int>();

### 协议类 G2C\_Supply\_Supply 返回 补兵

对应ID： G2C\_SUPPLY\_SUPPLY

[Desc("成员列表 <武将信息>")]

public Dictionary<long, CLS\_WarriorInfo> DictWarrior = new Dictionary<long, CLS\_WarriorInfo>();

### 协议类 C2G\_Supply\_SupplyAuto 请求 一键补兵

对应ID： C2G\_SUPPLY\_SUPPLYAUTO

[Desc("k=武将唯一ID")]

public List<long> ListWarrior = new List<long>();

### 协议类 G2C\_Supply\_SupplyAuto 返回 一键补兵

对应ID： G2C\_SUPPLY\_SUPPLYAUTO

[Desc("成员列表 <武将信息>")]

public Dictionary<long, CLS\_WarriorInfo> DictWarrior = new Dictionary<long, CLS\_WarriorInfo>();

## 战略

### 公共类 CLS\_StrategyCityInfoBase 城池战略信息基础

[Desc("所属势力唯一ID(0=无人占领)")]

public long GuildUid = 0;

[Desc("所属势力名")]

public string GuildName = "";

[Desc("势力所属国家ID")]

public int CountryId = 0;

[Desc("归属 EOwnership")]

public byte Ownership = 0;

[Desc("倒计时受保护")]

public TimeSpan TsCooldownProtected = new TimeSpan();

### 公共类 CLS\_StrategyCityInfo 战略城池信息

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("太守Id")]

public long LeaderPuid = 0;

[Desc("太守名字")]

public string LeaderName = "";

[Desc("繁荣度")]

public long Prosperity = 0;

[Desc("城池战略信息基础")]

public CLS\_StrategyCityInfoBase CityInfoBase = new CLS\_StrategyCityInfoBase();

[Desc("城池战略信息大城池")]

public CLS\_StrategyCityInfoMetro CityInfoMetro = new CLS\_StrategyCityInfoMetro();

### 协议类 C2G\_Strategy\_Map 请求 战略地图

对应ID： C2G\_STRATEGY\_MAP

### 协议类 G2C\_Strategy\_Map 返回 战略地图

对应ID： G2C\_STRATEGY\_MAP

[Desc("城池列表")]

public Dictionary<int, CLS\_StrategyCityInfo> DictCity = new Dictionary<int, CLS\_StrategyCityInfo>();

[Desc("大城池PVP综合信息")]

public CLS\_MetroPvpInfo MetroPvpInfo = new CLS\_MetroPvpInfo();

### 协议类 C2G\_Strategy\_CityInfo 请求 城池信息

对应ID： C2G\_STRATEGY\_CITYINFO

[Desc("城池唯一ID(配置ID)")]

public int Uid = 0;

### 协议类 G2C\_Strategy\_CityInfo 返回 城池信息

对应ID： G2C\_STRATEGY\_CITYINFO

[Desc("城池信息")]

public CLS\_StrategyCityInfo CityInfo = new CLS\_StrategyCityInfo();

### 公共类 CLS\_StrategyFightCampInfo 城池战斗守备营信息

[Desc("唯一ID(1-5)(1=主营)")]

public int Uid = 0;

[Desc("城墙耐久")]

public CLS\_CountInfo WallHp = new CLS\_CountInfo();

[Desc("攻击者列表")]

public List<string> ListLastAttacker = new List<string>();

### 公共类 CLS\_StrategyFightInfo 城池战斗信息

[Desc("唯一ID(配置ID)")]

public int Uid = 0;

[Desc("城池战略信息基础")]

public CLS\_StrategyCityInfoBase CityInfoBase = new CLS\_StrategyCityInfoBase();

[Desc("城池战斗守备营信息")]

public Dictionary<int, CLS\_StrategyFightCampInfo> DictCamp = new Dictionary<int, CLS\_StrategyFightCampInfo>();

### 协议类 C2G\_Strategy\_FightInfo 请求 城池战斗信息

对应ID： C2G\_STRATEGY\_FIGHTINFO

[Desc("城池唯一ID(配置ID)")]

public int Uid = 0;

### 协议类 G2C\_Strategy\_FightInfo 返回 城池战斗信息

对应ID： G2C\_STRATEGY\_FIGHTINFO

[Desc("城池战斗信息")]

public CLS\_StrategyFightInfo CityInfo = new CLS\_StrategyFightInfo();

### 协议类 G2C\_Strategy\_NotifyRefreshWallHp 通知 城墙血量刷新

对应ID： G2C\_STRATEGY\_NOTIFYREFRESHWALLHP

[Desc("城池ID")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("城墙耐久")]

public CLS\_CountInfo WallHp = new CLS\_CountInfo();

### 协议类 G2C\_Strategy\_NotifyRefreshCityInfo 通知 城池信息刷新

对应ID： G2C\_STRATEGY\_NOTIFYREFRESHCITYINFO

[Desc("城池信息")]

public CLS\_StrategyCityInfo CityInfo = new CLS\_StrategyCityInfo();

### 协议类 C2G\_Strategy\_MatchingCancel 请求 城池战斗匹配取消

对应ID： C2G\_STRATEGY\_MATCHINGCANCEL

### 协议类 G2C\_Strategy\_MatchingCancel 返回 城池战斗匹配取消

对应ID： G2C\_STRATEGY\_MATCHINGCANCEL

### 协议类 C2G\_Strategy\_Defense 请求 城池战斗防守

对应ID： C2G\_STRATEGY\_DEFENSE

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

### 协议类 G2C\_Strategy\_Defense 返回 城池战斗防守

对应ID： G2C\_STRATEGY\_DEFENSE

[Desc("匹配开始时间")]

public DateTime DtMatchingStart = DateTime.Now;

[Desc("匹配结束时间")]

public DateTime DtMatchingEnd = DateTime.Now;

### 协议类 C2G\_Strategy\_Attack 请求 城池战斗进攻

对应ID： C2G\_STRATEGY\_ATTACK

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

### 协议类 G2C\_Strategy\_Attack 返回 城池战斗进攻

对应ID： G2C\_STRATEGY\_ATTACK

[Desc("匹配开始时间")]

public DateTime DtMatchingStart = DateTime.Now;

[Desc("匹配结束时间")]

public DateTime DtMatchingEnd = DateTime.Now;

### 协议类 G2C\_Strategy\_NotifyMatchingEndDefense 通知 城池战斗匹配结束防守

对应ID： G2C\_STRATEGY\_NOTIFYMATCHINGENDDEFENSE

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("城墙血量当前值")]

public long WallHp = 0;

[Desc("城墙血量增加值")]

public long WallHpAdded = 0;

### 协议类 G2C\_Strategy\_NotifyMatchingAttackNpc 通知 城池战斗匹配攻击NPC

对应ID： G2C\_STRATEGY\_NOTIFYMATCHINGATTACKNPC

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("被攻击的NPC信息")]

public CLS\_CampNpc CampNpc = new CLS\_CampNpc();

### 协议类 G2C\_Strategy\_NotifyMatchingAttackWall 通知 城池战斗匹配攻击城墙

对应ID： G2C\_STRATEGY\_NOTIFYMATCHINGATTACKWALL

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("城墙等级")]

public int WallLevel = 0;

[Desc("城墙耐久")]

public CLS\_CountInfo WallHp = new CLS\_CountInfo();

### 公共类 CLS\_CampNpcMonster 守军NPC兵力

[Desc("ID(位置0-n)")]

public int MonsterUid = 0;

[Desc("怪物ID")]

public int MonsterId = 0;

[Desc("兵力")]

public long Hp = 0;

### 公共类 CLS\_CampNpc 守军NPC

[Desc("ID(1-n)")]

public int NpcUid = 0;

[Desc("关卡配置ID")]

public int StageId = 0;

[Desc("NPC部队兵力 k=位置0-n")]

public Dictionary<int, CLS\_CampNpcMonster> DictMonster = new Dictionary<int, CLS\_CampNpcMonster>();

### 协议类 C2G\_Strategy\_AttackNpcBalance 请求 城池战斗匹配攻击NPC结算

对应ID： C2G\_STRATEGY\_ATTACKNPCBALANCE

[Desc("战斗验证码")]

public long BattleCode = 0;

[Desc("战斗结果 EBattleResult")]

public int BattleResult = 0;

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("ID(1-n)")]

public int NpcUid = 0;

[Desc("NPC部队兵力 k=位置0-n")]

public Dictionary<int, CLS\_CampNpcMonster> DictMonster = new Dictionary<int, CLS\_CampNpcMonster>();

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHp = new Dictionary<long, int>();

### 协议类 G2C\_Strategy\_AttackNpcBalance 返回 城池战斗匹配攻击NPC结算

对应ID： G2C\_STRATEGY\_ATTACKNPCBALANCE

### 协议类 C2G\_Strategy\_AttackWallBalance 请求 城池战斗匹配攻击城墙结算

对应ID： C2G\_STRATEGY\_ATTACKWALLBALANCE

[Desc("战斗验证码")]

public long BattleCode = 0;

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("城墙血量损失")]

public long WallHpDamage = 0;

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHp = new Dictionary<long, int>();

### 协议类 G2C\_Strategy\_AttackWallBalance 返回 城池战斗匹配攻击城墙结算

对应ID： G2C\_STRATEGY\_ATTACKWALLBALANCE

[Desc("首次攻占城池奖励")]

public List<CLS\_AwardItem> ListAwardFirst = new List<CLS\_AwardItem>();

### 协议类 G2C\_Strategy\_NotifyMatchingAttackPlayer 通知 城池战斗匹配攻击玩家

对应ID： G2C\_STRATEGY\_NOTIFYMATCHINGATTACKPLAYER

[Desc("战场信息")]

public CLS\_BattleInfo BattleInfo = new CLS\_BattleInfo();

### 协议类 C2G\_Strategy\_AttackPlayerBalance 请求 城池战斗匹配攻击玩家结算

对应ID： C2G\_STRATEGY\_ATTACKPLAYERBALANCE

[Desc("战斗ID")]

public int BattleUid = 0;

[Desc("战斗结果 攻击方胜利=true 防守方胜利=false")]

public bool BattleResult = false;

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHpAtt = new Dictionary<long, int>();

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHpDef = new Dictionary<long, int>();

### 协议类 G2C\_Strategy\_AttackPlayerBalance 返回 城池战斗匹配攻击玩家结算

对应ID： G2C\_STRATEGY\_ATTACKPLAYERBALANCE

[Desc("战斗ID")]

public int BattleUid = 0;

[Desc("k=Puid v=结果 胜利=true 失败=false")]

public Dictionary<long, bool> DictBattleResult = new Dictionary<long, bool>();

## 大城池PVP

### 公共类 CLS\_MetroPvpInfo 大城池PVP综合信息

[Desc("大城池PVP综合状态")]

public EMetroPvpState MetroPvpState = new EMetroPvpState();

[Desc("开始时间")]

public DateTime DtMetroPvpStart = DateTime.Now;

[Desc("结束时间")]

public DateTime DtMetroPvpEnd = DateTime.Now;

### 公共类 CLS\_StrategyCityInfoMetro 城池战略信息大城池

[Desc("大城池状态")]

public EMetroCityStatus MetroCityStatus = new EMetroCityStatus();

[Desc("倒计时占领")]

public TimeSpan TsCooldownSeize = new TimeSpan();

### 协议类 G2C\_Metro\_NotifyRefreshInfo 通知 大城池PVP综合信息

对应ID： G2C\_METRO\_NOTIFYREFRESHINFO

[Desc("大城池PVP综合信息")]

public CLS\_MetroPvpInfo MetroPvpInfo = new CLS\_MetroPvpInfo();

### 公共类 CLS\_MetroTopScoreNode 积分榜单条

[Desc("排名")]

public int Rank = 0;

[Desc("势力唯一ID")]

public long GuildUid = 0;

[Desc("势力名")]

public string GuildName = "";

[Desc("积分")]

public long Score = 0;

### 协议类 C2G\_Metro\_TopScore 请求 郡城势力积分榜

对应ID： C2G\_METRO\_TOPSCORE

[Desc("城池唯一ID(配置ID)")]

public int Uid = 0;

### 协议类 G2C\_Metro\_TopScore 返回 郡城势力积分榜

对应ID： G2C\_METRO\_TOPSCORE

[Desc("城池唯一ID(配置ID)")]

public int Uid = 0;

[Desc("自己")]

public CLS\_MetroTopScoreNode ScoreMy = new CLS\_MetroTopScoreNode();

[Desc("前十")]

public List<CLS\_MetroTopScoreNode> ListScore = new List<CLS\_MetroTopScoreNode>();

### 协议类 C2G\_Metro\_Matching 请求 郡城匹配

对应ID： C2G\_METRO\_MATCHING

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

### 协议类 G2C\_Metro\_Matching 返回 郡城匹配

对应ID： G2C\_METRO\_MATCHING

[Desc("匹配开始时间")]

public DateTime DtMatchingStart = DateTime.Now;

[Desc("匹配结束时间")]

public DateTime DtMatchingEnd = DateTime.Now;

### 协议类 G2C\_Metro\_NotifyMatchingEnd 通知 郡城匹配结束

对应ID： G2C\_METRO\_NOTIFYMATCHINGEND

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

### 协议类 G2C\_Metro\_NotifyMatchingMetro 通知 郡城匹配成功

对应ID： G2C\_METRO\_NOTIFYMATCHINGMETRO

[Desc("战场信息")]

public CLS\_BattleInfo BattleInfo = new CLS\_BattleInfo();

### 协议类 C2G\_Metro\_AttackPlayerBalance 请求 郡城战斗结算

对应ID： C2G\_METRO\_ATTACKPLAYERBALANCE

[Desc("战斗ID")]

public int BattleUid = 0;

[Desc("战斗结果 攻击方胜利=true 防守方胜利=false")]

public bool BattleResult = false;

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHpAtt = new Dictionary<long, int>();

[Desc("武将消耗兵力")]

public Dictionary<long, int> DictExpendHpDef = new Dictionary<long, int>();

### 协议类 G2C\_Metro\_AttackPlayerBalance 返回 郡城战斗结算

对应ID： G2C\_METRO\_ATTACKPLAYERBALANCE

[Desc("战斗ID")]

public int BattleUid = 0;

[Desc("k=Puid v=结果 胜利=true 失败=false")]

public Dictionary<long, bool> DictBattleResult = new Dictionary<long, bool>();

## 即时对战

### 公共类 CLS\_BattlePlayer 即时对战玩家信息

[Desc("玩家唯一ID")]

public long Puid = 0;

[Desc("玩家昵称")]

public string Name = "";

[Desc("已准备")]

public bool IsReady = false;

[Desc("成员列表 <位置 武将战斗信息>")]

public Dictionary<int, CLS\_WarriorInfo> DictWarrior = new Dictionary<int, CLS\_WarriorInfo>();

### 公共类 CLS\_BattleInfo 战场信息

[Desc("战斗ID")]

public int BattleUid = 0;

[Desc("战场类型")]

public EBattleType BattleType = new EBattleType();

[Desc("已准备全部")]

public bool IsReadyAll = false;

[Desc("城池唯一ID(配置ID)")]

public int CityUid = 0;

[Desc("唯一ID(1-5)(1=主营)")]

public int CampUid = 0;

[Desc("攻击方")]

public CLS\_BattlePlayer BattlePlayerAtt = new CLS\_BattlePlayer();

[Desc("防守方")]

public CLS\_BattlePlayer BattlePlayerDef = new CLS\_BattlePlayer();

### 协议类 C2G\_Battle\_Demo 请求 测试

对应ID： C2G\_BATTLE\_DEMO

### 协议类 G2C\_Battle\_Demo 返回 测试

对应ID： G2C\_BATTLE\_DEMO

### 协议类 C2G\_Battle\_Ready 请求 准备

对应ID： C2G\_BATTLE\_READY

### 协议类 G2C\_Battle\_Ready 返回 准备

对应ID： G2C\_BATTLE\_READY

### 协议类 G2C\_Battle\_NotifyReady 通知 准备状态

对应ID： G2C\_BATTLE\_NOTIFYREADY

[Desc("已准备全部")]

public bool IsReadyAll = false;

### 公共类 CLS\_ForwardData 转发数据

[Desc("状态类型")]

public short Status = 0;

[Desc("状态数据")]

public Dictionary<string, string> DictStatusData = new Dictionary<string, string>();

[Desc("操作类型")]

public short Action = 0;

[Desc("操作数据")]

public Dictionary<string, string> DictActionData = new Dictionary<string, string>();

### 协议类 C2G\_Battle\_ForwardData 请求 转发数据

对应ID： C2G\_BATTLE\_FORWARDDATA

[Desc("武将唯一ID")]

public long WarriorUid = 0;

[Desc("转发数据")]

public CLS\_ForwardData ForwardData = new CLS\_ForwardData();

### 协议类 G2C\_Battle\_ForwardData 返回 转发数据

对应ID： G2C\_BATTLE\_FORWARDDATA

[Desc("玩家唯一ID")]

public long PlayerId = 0;

[Desc("武将唯一ID")]

public long WarriorUid = 0;

[Desc("转发数据")]

public CLS\_ForwardData ForwardData = new CLS\_ForwardData();

# 中转 27000

## 中转 27000

### 协议类 G2A\_Load\_SetLoad 集群服务器发送负载给中心服务器

对应ID： G2A\_LOAD\_SETLOAD

[Desc("服务器ID")]

public int Id = 0;

[Desc("服务器IP地址")]

public string Ip = "";

[Desc("服务器类型")]

public short LinkType = 0;

[Desc("服务器名字")]

public string Name = "";

[Desc("负载人数")]

public int Load = 0;

[Desc("进程ID")]

public int ProcessID = 0;

[Desc("使用CPU")]

public float Cpu = 0.0f;

[Desc("使用内存")]

public long Memory = 0;

[Desc("使用线程")]

public int Threads = 0;

# 管理工具

## 管理工具

### 协议类 B2T\_GM\_Start 集群服务器发送负载给中心服务器

对应ID： B2T\_GM\_START

### 协议类 B2T\_GM\_Login 请求 GM登陆

对应ID： B2T\_GM\_LOGIN

[Desc("账号")]

public string Account = "";

[Desc("密码")]

public string Password = "";

### 协议类 T2B\_GM\_Login 返回 GM登陆

对应ID： T2B\_GM\_LOGIN

### 协议类 B2G\_GM\_Cmd 请求 GM特殊功能

对应ID： B2G\_GM\_CMD

[Desc("指令")]

public string Cmd = "";

public List<string> Args = new List<string>();

### 协议类 G2B\_GM\_Cmd 请求 GM特殊功能

对应ID： G2B\_GM\_CMD

[Desc("指令")]

public string Cmd = "";

public List<string> Args = new List<string>();

### 协议类 T2B\_Mail\_GlobalList 请求 邮件列表

对应ID： T2B\_MAIL\_GLOBALLIST

### 协议类 B2T\_Mail\_GlobalList 返回 邮件列表

对应ID： B2T\_MAIL\_GLOBALLIST

[Desc("邮件列表")]

public List<CLS\_MailInfoDetails> ListMail = new List<CLS\_MailInfoDetails>();

### 公共类 CLS\_GmMailInfo GM给玩家发送邮件信息

[Desc("邮件ID")]

public long Id = 0;

public short TargetType = 0;

[Desc("接收者列表")]

public List<string> ListTarget = new List<string>();

[Desc("标题")]

public string Title = "";

[Desc("正文")]

public string Body = "";

[Desc("邮件发送者")]

public string SenderName = "";

[Desc("附件列表")]

public List<CLS\_AwardItem> ListAttachments = new List<CLS\_AwardItem>();

### 协议类 B2T\_GmMail\_Send 请求 GM给玩家发送邮件

对应ID： B2T\_GMMAIL\_SEND

[Desc("邮件信息")]

public CLS\_GmMailInfo MailInfo = new CLS\_GmMailInfo();

### 协议类 T2B\_GmMail\_Send 返回 GM给玩家发送邮件

对应ID： T2B\_GMMAIL\_SEND

[Desc("错误字符串")]

public string Msg = "";

### 协议类 B2T\_Edit\_Player 发送 修改玩家数据

对应ID： B2T\_EDIT\_PLAYER

public string IdType = "";

public string Id = "";

public string Mode = "";

public string Arg = "";

### 协议类 T2B\_Edit\_Player 返回 修改玩家数据

对应ID： T2B\_EDIT\_PLAYER

[Desc("错误字符串")]

public string Msg = "";

### 协议类 B2T\_Bank\_Conf 请求 长安夺宝配置

对应ID： B2T\_BANK\_CONF

[Desc("配置 （根据IsSeted判断是查看还是设置）")]

public CLS\_BankConf BankConf = new CLS\_BankConf();

### 协议类 T2B\_Bank\_Conf 返回 长安夺宝配置

对应ID： T2B\_BANK\_CONF

[Desc("配置")]

public CLS\_BankConf BankConf = new CLS\_BankConf();

### 协议类 B2T\_GM\_End 返回 长安夺宝配置

对应ID： B2T\_GM\_END