//--------------------服务器相关--------------------

//服务器列表请求

struct C2GM\_MsgServerListReqGMC2GMS

{

Type 2571

}

//服务器列表反馈

struct GM2C\_MsgServerList

{

Type 2505

uint16 usAmount; //数量

Server server[usAmount];

}

struct Server

{

uint32 idGameServer; //游服id

int16 name\_len; //游服名长度（上限32字节）

int8 szName[name\_len]; //游服名

int8 btState; //游服状态

}

//关服请求

struct C2GM\_MsgCloseServerGmcToGms

{

Type 2579

uint16 num; //数量

Server server[num];

Uint64 closetime; //关服时间

}

struct Server

{

uint32 idGameServer; //服务器id

}

//--------------------玩家信息相关--------------------

//--------------------公告相关--------------------

//循环公告列表请求

struct C2GM\_MsgAnnounceLoopListReqGMC2GMS

{

Type 2573

}

//循环公告列表反馈

struct GM2C\_MsgAnnounceLoopList

{

Type 2519

uint16 usAmount; //数量

Announce announce[usAmount];

}

struct Announce

{

uint32 idGameServer; //游服id

uint32 idAnnounce; //公告id

int16 content\_len; //内容（上限512字节）

int8 szContent[content\_len]; //内容

Uint64 u64SendTime; //发送时间

Uint64 u64EndTime; //结束时间

Uint64 u64Interval; //间隔时间

}

//定时公告列表请求

struct C2GM\_MsgAnnounceRegularListReqGMC2GMS

{

Type 2574

}

//定时公告列表反馈

struct GM2C\_MsgAnnounceRegularList

{

Type 2520

uint16 usAmount; //数量

Announce announce[usAmount];

}

struct Announce

{

uint32 idGameServer; //游服id

uint32 idAnnounce; //公告id

int16 content\_len; //内容（上限512字节）

int8 szContent[content\_len]; //内容

Uint64 u64SendTime; //发送时间

}

//发布循环公告请求

struct C2GM\_MsgAnnounceLoop

{

Type 2521

uint32 idGameServer; //游服id

int16 content\_len; //内容（上限512字节）

int8 szContent[content\_len]; //内容

Uint64 u64SendTime; //发送时间

Uint64 u64EndTime; //结束时间

Uint64 u64Interval; //间隔时间

}

//发布定时公告请求

struct C2GM\_MsgAnnounceRegular

{

Type 2522

uint32 idGameServer; //游服id

int16 content\_len; //内容（上限512字节）

int8 szContent[content\_len]; //内容

Uint64 u64SendTime; //发送时间

}

//发布及时公告请求

struct C2GM\_MsgAnnounceInstant

{

Type 2523

uint32 idGameServer; //游服id

int16 content\_len; //内容（上限512字节）

int8 szContent[content\_len]; //内容

}

//删除循环公告请求

struct C2GM\_MsgAnnounceLoopDel

{

Type 2525

uint32 idGameServer; //游服id

uint32 idAnnounce; //公告id

}

//删除循环公告反馈

struct GM2C\_MsgAnnounceLoopDelRet

{

Type 2526

uint32 idGameServer; //游服id

uint32 unRet; //返回码（0成功，1失败）

}

//删除定时公告请求

struct C2GM\_MsgAnnounceRegularDel

{

Type 2527

uint32 idGameServer; //游服id

uint32 idAnnounce; //公告id

}

//删除定时公告反馈

struct GM2C\_MsgAnnounceRegularDelRet

{

Type 2528

uint32 idGameServer; //游服id

uint32 unRet; //返回码（0成功，1失败）

}

//--------------------玩家操作相关--------------------

//踢人请求

struct C2GM\_MsgKickUserGMC2GMS

{

Type 2529

uint32 idGameServer; //游服id

uint32 idUser; //角色id

}

//踢人结果

struct GM2C\_MsgKickUserRetGMS2GMC

{

Type 2532

uint32 idGameServer; //游服id

uint32 unRet; //返回码（0成功，1失败）

}

//禁言玩家列表请求

struct C2GM\_MsgUserSilenceListReq

{

Type 2515

uint32 idGameServer; //游服id

}

//禁言玩家列表反馈

struct GM2C\_MsgUserSilenceList

{

Type 2516

uint32 idGameServer; //游服id

uint16 usAmount; //数量

Name name[usAmount];

}

struct Name

{

int16 name\_len; //名字产度（上限64字节）

int8 szName[name\_len]; //名字

}

//禁言请求

struct C2GM\_MsgSilenceUserGMC2GMS

{

Type 2533

uint32 idGameServer; //游服id

int16 name\_len; //名字产度（上限64字节）

int8 szName[name\_len]; //名字

Bool bSilence; //是否禁言（true禁言，flase解禁）

}

//禁言反馈

struct GM2C\_MsgSilenceUserRetGMS2GMC

{

Type 2536

uint32 idGameServer; //游服id

uint32 unRet; //返回码（0成功，1失败）

}

struct Character

{

uint32 idCharacter; //角色id

}

//发送邮件请求

struct C2GM\_MsgGMMailReqGMC2GMS

{

Type 2539

uint32 idGameServer; //游服id

uint16 usCharacterAmount; //角色数量（0表示全服发放）

Character character[usCharacterAmount];

Uint32 money;

Uint32 emoney;

int16 content\_len; //邮件正文长度（上限200字节）

int8 szContent[content\_len]; //邮件正文

uint16 num; //邮件附件数量

MailThing mailthing[num]; //邮件附件

}

struct Character

{

uint32 idCharacter; //角色id

}

//附加属性信息

Struct AttrInfo

{

uint32 attrType;

//属性类型（0无，1攻击，2防御，3生命，4命中，5闪避，6暴击，7韧性，11攻击百分比，12防御百分比，13生命百分比，14命中百分比，15闪避百分比，16暴击百分比，17韧性百分比）

}

//装备信息

Struct EquipData

{

uint32 equipid; //装备id

uint32 num;//数量

bool isBind; //是否绑定（0绑定，1未绑定）

uint16 realm;

uint16 stage;

uint32 equiplev;

uint16 num; //附加属性数量

AttrInfo info[num]; //属性

}

//物品信息

Struct ItemData

{

uint32 itemid; //物品id

uint32 num; //物品数量

bool isBind; //是否绑定（0绑定，1未绑定）

}

//邮件附件

Struct MailThing

{

uint16 usThingType;

//附件类型（0物品，1装备）

union

{

ItemData itemdata; //物品信息（附件类型8）

EquipData equipdata; //装备信息（附件类型9）

}

}

//传送请求

struct C2GM\_MsgGMTeleportGMC2GMS

{

Type 2541

uint32 idGameServer; //游服id

uint16 usAmount; //角色数量

Character character[usAmount];

uint32 idScene; //场景id

Float fPosX; //坐标x

Float fPosY; //坐标y

}

struct Character

{

uint32 idCharacter; //角色id

}

//全服发放请求

struct C2GM\_MsgGMGiftAllGMC2GMS

{

Type 2565

uint32 idGameServer; //游服id

uint32 idReward; //奖励id

uint32 unNumber; //数量

}

//锁定帐号请求

struct C2GM\_MsgLockAccountGmcToGms

{

Type 2581

uint32 idGameServer; //服务器id

int16 account\_len; //需要锁定的帐号长度（上限64字节）

int8 szAccount[account\_len]; //需要锁定的帐号

}

//解锁帐号请求

struct C2GM\_MsgUnLockAccountGmcToGms

{

Type 2583

uint32 idGameServer; //服务器id

int16 account\_len; //需要解锁的帐号长度（上限64字节）

int8 szAccount[account\_len]; //需要解锁的帐号

}

//锁定账号列表查询

struct MsgGMLockAccountListGMC2GMSReq

{

Type 8518

uint32 idGameServer; //服务器id

}

//锁定账号列表返回

struct MsgGMLockAccountListGMS2GMCRet

{

Type 8521

uint32 idGameServer; //服务器id

uint16 usLockAccountNumber; //锁定账号数量

Account account[usLockAccountNumber] //账号

}

struct Account

{

int16 account\_len; //锁定账号长度（上限64字节）

int8 szLockAccount[account\_len]; //锁定账号

}

//通用操作返回

struct GM2C\_MsgGMCommonRet

{

Type 8511

uint32 idGameServer; //服务器id

uint32 unRet; //返回码（0成功，1失败）

}

//GM充值请求（需要角色在线）

struct GM2C\_MsgGMChargeGMC2GMSReq

{

Type 8546

uint32 idGameServer; //服务器id

uint32 idCharacter; //角色id

uint32 unEmoney; //充值金额

}

//--------------------活动操作相关--------------------

struct C2GM\_MsgGMMultipleexpSetActivityGMC2GMSReq//设置活动

{

Type 26251

uint32 idGameServer;

uint32 multiple;

uint64 starttime;

uint64 endtime;

string desc;//长度限定254字符

}

struct C2GM\_MsgGMMultipleexpCloseActivityGMC2GMSReq,//关闭活动

{

Type 26253

uint32 idGameServer; //服务器id

}

struct C2GM\_MsgGMMultipleexpCheckInfoGMC2GMSReq,//活动信息查询

{

Type 26255

uint32 idGameServer; //服务器id（为0时，表示查询所有服务器

}

struct GM2C\_MsgGMMultipleexpCheckInfoGMS2GMCRet//活动信息查询返回

{

Type 26258

uint32 idGameServer;

uint32 activityStatus;

uint32 multiple;

uint64 starttime;

uint64 endtime;

}