## 通信方式

WebSocket

## 数据包头格式

Int 数据转换标志

Short 客户端类型(Client Type)

Long Token

Short 协议号(Opreation Code)

Int 服务器状态码(Server Status Code)

## 常规请求通信命令

### 用户登录 OperCode=1

请求包体格式

{

string 用户名

string 密码

string 设备ID

string Mac地址

string 预留字段

string 预留字段

string 预留字段

byte 操作系统（1.ios 2.android）

string 渠道名称

}

返回包体格式

{

int 登录结果(返回值不为0没有下述数据)

byte 玩家是否已创建角色

string 游戏逻辑服务器地址(Game Server Address)

}

### 用户注册OperCode=2

请求包体格式

{

string 用户名

string 密码

string 设备ID

string Mac地址

string 预留字段

string 预留字段

string 预留字段

byte操作系统（1.ios 2.android）

string 渠道名称

}

返回包体格式

{

int 登录结果

}

### 获取玩家基本属性OperCode=3

请求包体格式

{

}

返回包体格式

{

int 操作结果

guid 玩家id

string 角色名

int 玩家等级

int 经验

int 生命

int 防御力

int cost值上限

int 战斗力

int SP点数

int 当前体力值

int 体力上限

int 卡包上限

int 好友上限

}

### 获取卡组及卡牌信息OperCode=4

请求包体格式

{

}

返回包体格式

{

int 操作结果

int卡组数

卡组列表

{

guid 卡组ID

byte 卡组类型 (PVE : 1 , PVP : 2)

string卡组名称(目前有两个卡组PVE，PVP)

int卡组内的卡牌数

卡组内卡牌列表

{

guid 卡牌ID

}

}

int 卡牌数

卡牌列表

{

guid 卡牌ID

string 模板ID

int 卡牌等级

int 经验

int 主技能等级

int子技能等级

guid 父卡ID(如果没有父卡，ID是：Guid.Empty)

int 子卡数

子卡ID

{

guid 子卡ID

}

}

}

### 加入副本OperCode=5

请求包体格式

{

byte 副本类型(0:普通 1:好友 2:附近)

string 地图模板ID

}

返回包体格式

{

int 操作结果

guid 房间ID

byte 是否乱入（0：不是 1：是）

int场上人数

场上玩家列表

{

[房间内玩家信息](#房间内玩家信息)

}

}

### 是否复活OperCode=6

请求包体格式

{

byte 是否复活（0 ：不复活 1：复活）

}

返回包体格式

{

int 操作结果

}

### bet OperCode=7

待定

### SP 技能释放OperCode=8

请求包体格式

{

}

返回包体格式

{

}

### 出牌OperCode=9

请求包体格式

{

卡牌ID

Int 当前服务器操作计数

}

返回包体格式

{

int 操作结果

}

### 回合结束通知OperCode=10

请求包体格式

{

Int 当前服务器操作计数

}

返回包体格式

{

int 操作结果

}

## 推送通信命令

### 1001. 回合准备阶段有玩家加入房间(自己乱入上阵则给自己发送1004)

{

guid玩家ID

string玩家角色名

byte是否是NPC(0:不是 1：是)

int 玩家等级

int 当前血量

int血量上限

byte 是否死亡(0:未死亡)

int 防御增加值

int 攻击增加值

int 素早

int buff数

buff列表

{

string buff模板ID

}

guid 替换的玩家ID（若无则为GUID\_EMPTY）

}

### 1003. 发送新出现的怪堆数据

{

int 怪物数量

怪物列表

{

guid 怪物ID

string 怪物模板ID

int 怪物等级

byte 怪物类型(0 普通怪 1 主怪2boss)

int 怪物顺序

int 当前血量

int 血量上限

int buff数

buff列表

{

string buff 模板ID

}

}

int扣除的体力值

}

### 1004. 回合开始阶段校验数据(针对乱入玩家与回合滞后玩家)

{

int 玩家数量

玩家列表

{

[房间内玩家信息](#房间内玩家信息)

}

int 怪物数量

怪物列表

{

[怪堆数据](#怪堆数据)

}

int 体力

int 经验

int 金币

int 卡牌数

int sp值

}

### 1006. 出牌阶段指定玩家出牌(推送给房间内所有玩家)

{

Guid 玩家Id

}

### 1007. 出牌阶段推送牌堆给指定玩家摸牌(只推送给单一玩家)

{

Int 牌数量

牌列表

{

Guid 卡牌Id

Byte 扑克对应的数字

Byte 对应的出牌索引

}

Guid 超时出的卡牌Id

Int 当前服务器操作计数

}

### 1008. 出牌阶段玩家出牌

{

Guid 玩家Id

String 卡牌模板Id

Byte 卡牌对应的数字

Int 技能等级

Guid 卡牌Id

}

### 1009. 攻击数据

{

byte 本方牌型 ()

有效的卡牌ID列表

{

Guid 卡牌ID

}

攻击动作列表

{

guid 攻击者人物ID

short 技能ID

关联的卡牌列表(合体技与超合体技)

{

guid 卡牌ID

short 技能ID

}

己方新增buff 列表

{

guid 玩家ID

short buffID

}

受击动作列表

{

guid 受击者人物ID

受伤害掉血值

{

byte 受击状态(0:正常 1:MISS)

int 掉血量

}

}

敌方新增buff 列表

{

guid 玩家ID

short buffID

}

}

结算列表

{

guid 人物ID

int 当前hp

byte 是否死亡

int 掉落经验

int 掉落金钱

string 掉落卡牌模板ID

全部buff 列表

{

short buffID

}

}

byte 本波怪是否清空

byte 副本是否完成

Int 当前服务器操作计数

}