

- 通用协议
- 通用结构
 - 名词解释(或游戏文档)
 - 协议文件名
 - 相关配置
 - ErnnRoomConfig.xls[房间表]
 - ErnnRoomStockConfig.xls[房间库存表]
 - 游戏通用结构
 - 通知类型
 - 牌型
 - 卡数据
 - 输赢信息
 - 玩家信息
 - 闲家下注
 - 协议列表
 - 进入房间
 - 请求场景信息
 - 准备
 - 玩家数据
 - 开牌
 - 选倍
 - 发送扑克
 - 抢庄
 - 游戏结束
 - 玩家通知消息
 - 协议ID枚举

通用协议

通用结构

名词解释(或游戏文档)

暂无

协议文件名

game_ernn_protocol.proto
game_ernn_def.proto

相关配置

ErnnRoomConfig.xls[房间表]

字段名	描述	备注
RoomID	房间id	
RoomName	房间名	
GoldCondition	金币条件	
BaseGold	底分	
BotMinGold	机器人最小金币	
BotMaxGold	机器人最大金币	
BroadcastGold	广播金币	

ErnnRoomStockConfig.xls[房间库存表]

字段名	描述	备注
RoomID	房间id	
RoomName	房间名	
BaseStock	基础库存	
InitStock	初始库存	
StockStage	库存阶段（库存在某个阶段对于不同收放分）	
WeakKill	弱杀概率	
StrongKill	强杀概率	
BrightWater	明抽	
MaxStock	最大库存	
RelushTime	刷新时间	

游戏通用结构

通知类型

```
enum e_notify_type {
    e_nt_table_dismiss = 1;    // 桌子解散
    e_nt_gold_not_enough = 2;  // 钱不够
    e_nt_shutdown = 3;        // 关服
    e_nt_assembling = 5;       // 组桌中
    e_nt_player_leave = 6;     // 对家离开
    e_nt_kill_ok = 7;         // gm杀分设定完成
}
```

牌型

```
#define CARD_VALUE_WX          16    // 五小牛
#define CARD_VALUE_ZD          15    // 炸弹牛
#define CARD_VALUE_JN          14    // 金牛
#define CARD_VALUE_YN          13    // 银牛
#define CARD_VALUE_NIUNIUI     11    // 牛牛
#define CARD_VALUE_NINE        10    // 牛9
#define CARD_VALUE_EIGHT       9     // 牛8
#define CARD_VALUE_SEVEN       8     // 牛7
#define CARD_VALUE_SIX         7     // 牛6
#define CARD_VALUE_FIVE        6     // 牛5
#define CARD_VALUE_FOUR        5     // 牛4
#define CARD_VALUE_THREE       4     // 牛3
#define CARD_VALUE_TWO         3     // 牛2
#define CARD_VALUE_ONE         2     // 牛1
#define CARD_VALUE_ZERO        1     // 无牛
```

卡数据

参数名	类型	描述	参数ID	默认值
card	repeated int32	[3+2]扑克列表	1	
card_type	int32	牌型	2	

输赢信息

参数名	类型	描述	参数ID	默认值
chair_id	int32	坐位号	1	
game_score	int32	输赢金币	4	

玩家信息

参数名	类型	描述	参数ID	默认值
-----	----	----	------	-----

参数名	类型	描述	参数ID	默认值
player_id	int32	玩家ID	1	
player_name	string	玩家昵称	2	
head_frame	int32	头像框	3	
head_custom	string	玩家头像	4	
player_gold	int64	玩家金币	5	
player_sex	int32	玩家性别	6	
vip_level	int32	Vip等级	7	

闲家下注

参数名	类型	描述	参数ID	默认值
chair_id	int32	坐位号	1	
bets	int32	1倍、5倍、10倍、20倍、40倍	2	

协议列表

进入房间

说明:现有架构这条协议是不会用到的,进入房间走的是logic的enter_game，进入游戏同时进入房间。
前端请求协议:packetc2l_enter_room

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_c2l_enter_room = 10009

后端返回协议:packetl2c_enter_room_result

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_l2c_enter_room_result = 15009
result	msg_type_def.e_msg_result_def	结果状态		

前端逻辑判断

有进入金币限制的游戏,判断玩家金币是否满足,百人游戏,通常不限制

请求场景信息

说明:进入桌子跟断线重连情况下调用,获取整个场景的数据

前端请求协议:packetc2l_get_scene_info

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_c2l_get_scene_info = 10002

后端返回协议:packetl2c_scene_info_result

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_l2c_scene_info_result = 15002
room_type	int32	房间Id	2	
decide_banker	int32	正在叫庄	4	
other_bets	repeated msg_other_bets	闲家下注	5	
cards	repeated cards_data	[2]2家扑克列表	6	
open_card	bool	[2]开牌	7	
game_end	packetl2c_game_end	结算	8	
trustee	int32	0不托,1倍、5倍、10倍、20倍、40倍	9	
status_remain_sec	int32	桌子状态剩余秒数	10	
status_turn_sec	int32	桌子状态秒数	11	

准备

说明:前端必须首先发准备消息

前端逻辑判断

前端请求协议:packetc2l_ready

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_c2l_ready = 10011
change_table	bool	是否换桌	2	

玩家数据

说明:当组桌成功，后端会告知所有玩家
后端返回协议:**packetl2c_send_all_playerinfos**

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_l2c_send_all_playerinfos = 15017
player_info	repeated msg_player_info	[2] 玩家信息	2	
banker_user	int32	叫庄用户	3	

开牌

说明:游戏中玩家得开牌请求及通知
前端请求协议:**packetc2l_open_card**

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_c2l_open_card = 10014

后端返回协议:**packetl2c_open_card_notify**

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_l2c_open_card_notify = 15019
current_user	int32	开牌用户	2	

选倍

说明:游戏中玩家得选倍请求及通知
前端请求协议:**packetc2l_bet_req**

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_c2l_bet_req = 10013

参数名	类型	描述	参数ID	默认值
bets	int32	1倍、5倍、10倍、20倍、40倍	2	

后端返回协议:packetl2c_bet_notify

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_l2c_bet_notify = 15014
current_user	int32	当前用户	2	
bets	int32	1倍、5倍、10倍、20倍、40倍	3	

发送扑克

说明:指定用户取牌

后端返回协议:packetl2c_send_card

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_l2c_send_card = 15020
cards	repeated cards_data	[2]2家扑克列表	2	

抢庄

说明:游戏中玩家得抢庄请求及通知

前端请求协议:packetc2l_banker_req

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_c2l_banker_req = 10012
grab_banker	bool	是否叫庄	2	

后端返回协议:packetl2c_banker_notify

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_l2c_banker_notify = 15021

参数名	类型	描述	参数ID	默认值
grab_user	int32	当前用户	2	
grab_banker	bool	是否叫庄	3	

游戏结束

后端返回协议:**packetl2c_game_end**

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_l2c_game_end = 15016
cards	repeated msg_card_info	[2]输赢金币	2	

玩家通知消息

说明:桌子解散，钱不够等通知消息的发送
后端返回协议:**packetl2c_game_notify**

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id	1	e_mst_l2c_game_notify = 15018
notify_type	e_notify_type	通知类型	2	
notify_param	int32	通知参数	3	

协议ID枚举


```
enum e_server_msg_type {
```

```
    //客户端到服务端-----
```

```
    e_mst_start_c2l = 10000;
```

```
    e_mst_c2l_check_state = 10001;  
    e_mst_c2l_get_scene_info = 10002;  
    e_mst_c2l_enter_room = 10009;  
    e_mst_c2l_leave_room = 10010;  
    e_mst_c2l_ready = 10011;  
    e_mst_c2l_banker_req = 10012;  
    e_mst_c2l_bet_req = 10013;  
    e_mst_c2l_open_card = 10014;  
    e_mst_c2l_gm_list = 10015;  
    e_mst_c2l_gm_kill = 10016;  
    e_mst_c2l_user_trustee = 10007;
```

```
    // 服务端到客户端-----
```

```
    e_mst_start_l2c = 15000;
```

```
    e_mst_l2c_check_state_result = 15001;  
    e_mst_l2c_scene_info_result = 15002;  
    e_mst_l2c_user_trustee = 15008;  
    e_mst_l2c_enter_room_result = 15009;  
    e_mst_l2c_leave_room_result = 15010;  
    e_mst_l2c_bet_notify = 15014;  
    e_mst_l2c_game_end = 15016;  
    e_mst_l2c_send_all_playerinfos = 15017;  
    e_mst_l2c_game_notify = 15018;  
    e_mst_l2c_open_card_notify = 15019;  
    e_mst_l2c_send_card = 15020;  
    e_mst_l2c_banker_notify = 15021;  
    e_mst_l2c_gm_list = 15024;
```

```
    e_mst_clend_index = 20000;
```

```
}
```