

- 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
 - [LandlordsFreeRoomConfig.xls](#)[房间表]
 - [LandlordsRoomStockConfig.xls](#)[房间库存表]
 - 游戏通用结构
 - 通知类型
 - 玩家信息
 - [room](#)场景信息
 - [score](#)场景信息
 - 加倍场景信息
 - [playing](#)场景信息
 - 协议列表
 - 进入房间
 - 请求场景信息
 - 准备
 - 玩家数据
 - 发牌
 - 叫分
 - 游戏开始
 - 用户加倍
 - 不洗牌场游戏开始
 - 出牌数据包
 - 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_land_protocol.proto
game_land_def.proto

相关配置

LandlordsFreeRoomConfig.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	

LandlordsRoomStockConfig.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
}
```

玩家信息

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	

room场景信息

参数名	类型	描述	参数ID	默认值
ready	bool	是否准备	1	

score场景信息

参数名	类型	描述	参数ID	默认值
current_user	int32	当前玩家	1	
remain_seconds	int32	剩余秒数	2	
score_info	repeated int32	[3] 叫分信息,0(不叫),1,2,4,8,255(没轮到),88(不qiang)	3	

参数名	类型	描述	参数ID	默认值
user_trustee	repeated bool	[3]托管信息	4	
cards_info	repeated int32	[17]扑克列表	5	

加倍场景信息

参数名	类型	描述	参数ID	默认值
remain_seconds	int32	剩余秒数	1	
user_trustee	repeated bool	[3]托管信息	2	
cards_info	repeated int32	[17]扑克列表	3	
land_user	int32	地主玩家	4	
land_score	int32	地主分数	5	
back_card	repeated int32	[3]底牌扑克	6	

playing场景信息

参数名	类型	描述	参数ID	默认值
land_chair_id	int32	地主玩家	1	
land_score	int32	地主分数 1,2,4,8	2	
current_user	int32	当前玩家	3	
remain_seconds	int32	剩余秒数	4	
user_trustee	repeated bool	[3]托管信息	5	
remain_card_cnt	int32	[3]剩余扑克数地主分数	6	
back_card	repeated int32	[3]底牌扑克	7	
cards_info	repeated int32	[20]扑克列表	8	
turn_cards1	repeated int32	[chair1]出牌列表 [空=不出,255=没轮到....]	9	
turn_cards2	repeated int32	[chair2]出牌列表	10	
turn_cards3	repeated int32	[chair3]出牌列表	11	

参数名	类型	描述	参数ID	默认值
new_turn	bool	新一轮	12	
bomb_time	int32	炸弹倍数	13	
out_cards	int32 out_cards	已出牌列表	14	
score_times	int32	分数倍数	15	

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

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
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	
scene_info1	msg_room_scene_info	room场景信息	3	
scene_info2	msg_score_scene_info	score场景信息	4	
scene_info3	msg_playing_scene_info	playing场景信息	5	
info_type	int32	哪种场景信息	6	
scene_info4	packetl2c_game_end	结算信息	7	
land_chair_id4	int32	地主玩家	8	
land_score4	int32	地主分数	9	
score_times4	int32	分数倍数	10	
task_id	int32	分数倍数	11	
times	repeated int32	[3]加倍阶段 [0: 未选择, 1: 不加倍, 2:加倍]	12	
scene_info5	msg_times_scene_info	加倍场景信息	13	
play_cnt	int32	当天已玩次数	14	

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

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	[3] 玩家信息	2	

发牌

说明:当组桌成功，后端会告知所有玩家牌数据
后端返回协议:**packetl2c_send_all_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_all_card = 15011
current_user	int32	当前玩家	2	
user_trustee	bool	[3]托管信息	3	
cards_info	repeated int32	[17]扑克列表	4	
cards_info2	repeated int32	[17]扑克列表, 机器人用	5	
cards_info3	repeated int32	[17]扑克列表, 机器人用	6	
back_card	repeated int32	[3]底牌扑克	7	
room_id	int32	房间Id	9	
task_id	int32	任务Id	10	
play_cnt	int32	当天已玩次数	12	
其它字段		机器人用		

叫分

说明:玩家叫地主，当3家都不叫时后端会重复发牌packetl2c_send_all_card
前端请求协议:packetc2l_land_score

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_land_score = 10012
score	bool	叫，不叫	2	

后端返回协议:packetl2c_land_score

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_land_score = 15012
land_user	int32	叫分玩家	2	
current_user	int32	当前玩家	3	
land_score	int32	上次叫分	4	
current_score	int32	当前叫分	5	

游戏开始

说明:后端叫分完成后主动发，此后普通场转入游戏状态，不洗牌场转入加倍状态
后端返回协议:packetl2c_game_start

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_start = 15013
land_user	int32	地主玩家	2	
land_score	int32	地主分数	3	
back_card	repeated int32	[3]底牌扑克	4	
score_times	int32	分数倍数，普通场为1， 不洗牌场跟据3张底牌是否有大小王，对子等为1,2,3...	5	
need_times	bool	不洗牌场为true	7	

用户加倍

说明:只有不洗牌场有效
前端请求协议:packetc2l_times

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_times = 10014
times	int32	是否加倍 [1:不加倍, 2:加倍]	2	

后端返回协议:packetl2c_times_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_times_notify = 15019
chair_idx	int32	玩家坐位号	2	
times	int32	是否加倍 [1:不加倍, 2:加倍]	3	

不洗牌场游戏开始

说明:不洗牌场，当3家都加倍后后端会主动发，此后不洗牌场转入游戏状态
后端返回协议:packetl2c_really_game_start

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_really_game_start = 15020

出牌数据包

说明:游戏中玩家得出牌请求及通知, 后端返回分成出牌和不出牌发送
前端请求协议:packetc2l_out_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_out_card = 10013
cards	repeated int32	出牌数据	2	

后端返回协议:packetl2c_out_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_out_card = 15014
out_card_user	int32	出牌玩家	2	
current_user	int32	当前玩家	3	
bomb_time	int32	炸弹倍数	4	
cards	int32	出牌数据	5	

后端返回协议:packetl2c_pass_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_pass_card = 15015
pass_card_user	int32	不出玩家	2	
current_user	int32	当前玩家	3	
new_turn	int32	新一轮， 当新一轮开始时， 可以任意出牌， 否则每次出牌必须压住上一个出牌	4	

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
bomb_time	repeated int32	每家的总倍数，包括 炸弹， 春天，用户加倍等等	2	
game_score	repeated int32	每家的输赢金币	3	
leave_game	repeated bool	逃跑标记	4	

参数名	类型	描述	参数ID	默认值
spring	bool	春天标记	5	
cards1	repeated int32	玩家1的牌数据	6	
cards2	repeated int32	玩家2的牌数据	7	
cards3	repeated int32	玩家3的牌数据	8	
limited	repeated bool	每家的获利限定	9	
task_ok	int32	任务完成标记（-1，没完成，0,1,2，完成的玩家坐位号）	10	
low_limited	repeated bool	每家的输钱限定	11	

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	

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_user_trustee = 10008;
    e_mst_c2l_enter_room = 10009;
    e_mst_c2l_leave_room = 10010;
    e_mst_c2l_ready = 10011;
    e_mst_c2l_land_score = 10012;
    e_mst_c2l_out_card = 10013;
    e_mst_c2l_times = 10014;
    e_mst_c2l_report = 10015;

    // 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_send_all_card = 15011;
    e_mst_l2c_land_score = 15012;
    e_mst_l2c_game_start = 15013;
    e_mst_l2c_out_card = 15014;
    e_mst_l2c_pass_card = 15015;
    e_mst_l2c_game_end = 15016;
    e_mst_l2c_send_all_playerinfos = 15017;
    e_mst_l2c_game_notify = 15018;
    e_mst_l2c_times_notify = 15019;
    e_mst_l2c_really_game_start = 15020;
    e_mst_l2c_report_res = 15021;

    e_mst_clend_index = 20000;

}
```