德州扑克

协议文件名

game_texaspoker_protocol.proto

game_texaspoker_def.proto

游戏通用结构

房间信息

msg_room_info

参数名	类型	描述	参数ID	默认值
roomid	int32	房间Id	1	

玩家信息

PlayerInfo

参数名	类型	描述	参数ID	默认值
playerId	int32	玩家ID	1	
nickName	string	昵称	2	
curChip	int64	当前筹码数量	3	
hasBet	int64	已下注金额	4	
hasBetCur	int64	已下注金额（本阶段内）	5	
state	int32	当前所处状态 EDeskPlayerState枚举	6	
seat	int32	玩家所在座位号	7	
declareResult	int32	玩家表态结果 EDeclare	8	
headFrame	int32	头像框	9	
headCustom	string	玩家头像	10	
sex	int32	玩家性别	11	
vipLevel	int32	Vip等级	12	
supply	bool	是否开启自动补充筹码到最大值	13	
supply_done	bool	是否触发自动补充	14	

玩家下注信息

PlayerBetInfo

参数名	类型	描述	参数ID	默认值
playerId	int32	玩家ID	1	
hasBet	int64	已下注金额	2	

牌信息

PokerInfo

参数名	类型	描述	参数ID	默认值
flower	int32	花色	1	
value	int32	面值	2	

游戏信息

GameInfo

参数名	类型	描述	参数ID	默认值
bankerSeat	int32	庄家所在座位	1	
smallBlindSeat	int32	小盲注所在座位	2	
bigBlindSeat	int32	大盲注所在座位	3	

结算玩家信息

BalancePlayerInfo

参数名	类型	描述	参数ID	默认值
playerId	int32	玩家ID	1	
seat	int32	玩家所在座位号，	2	
backPoker	PokerInfo[]	玩家底牌	3	
selPoker	PokerInfo[]	最终的5张牌（目前没有使用该字段）	4	
pokerType	int64	最终5张牌的牌型 EPokerType	5	
winReward	int32	赢得的奖励	6	
isAbandon	bool	是否弃牌	7	
isWinner	bool	是否赢家	8	
isShowPoker	bool	是否亮牌	9	
rank	string	排名	10	

协议列表

检测游戏状态 //重连之后判断是否在桌子中

说明：用于断线重连，登录游戏后发送，如果返回在房间中，则走重连逻辑，直接进房间

前端请求协议:packetc2l_check_state

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_check_state 10009

后端返回协议:packetc2l_check_state_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_check_state_result 15014
is_intable	bool	是否在桌子中	2	

获得房间信息

说明:给前端用来请求房间列表信息（目前前端似乎没用调用，前端目前是使用配置展示房间列表）

前端请求协议:packetc2l_get_room_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_room_info 10001

后端返回协议:packetl2c_get_room_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_get_room_info_result 15001
room_list	msg_room_info[]	房间列表	2	

进入桌子（暂时不使用，游戏中使用通用进房间逻辑）

说明：暂时不使用，游戏中使用通用进房间逻辑

前端请求协议:packetc2l_join_table

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_join_table 10002
roomId	int32	房间ID	2	

后端返回协议:packetl2c_join_table_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_join_table_result 15002
result	int32	房间列表	2	1.成功 非1.失败

离开桌子（暂时不使用，游戏中使用通用出房间逻辑）

说明：暂时不使用，游戏中使用通用出房间逻辑

前端请求协议:packetc2l_leave_table

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_leave_table 10003

后端返回协议:packetl2c_leave_table_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_leave_table_result 15003
result	int32	房间列表	2	1.成功 非1.失败
playerGold	int32	玩家金币	3	

游戏事件通知(服务器->客户端)

说明：服务端主动推送给玩家的消息目前用于以下几处

1.开始表态 2.表态成功 3.发牌 4.玩家加入 5.玩家离开 6.玩家从座位上站起

后端返回协议:packetl2c_game_event_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_event_notify 15005
event	int32	事件类型 EGameEvent	2	
playerInfo	PlayerInfo	玩家信息	3	
commonPoker	PokerInfo[]	公共牌信息	4	
curRewardPool	int64	当前底池	5	
addBetBaseValue	int64	加注时基本值	6	
curBet	int64	本轮底注	7	
playerBetInfo	PlayerBetInfo[]	玩家历史下注（包括弃牌离场玩家）	8	

请求桌子内场景信息

说明：客户端刚进房间的时候请求，用于客户端初始化房间

前端请求协议:packetc2l_get_table_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_table_scene_info 10004

后端返回协议:packetl2c_get_table_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_get_table_scene_info_result 15004
roomId	int32	房间id	2	
tableId	int32	桌子id	3	
gameState	int32	当前游戏所处状态 EDeskState	4	
PlayerInfo	playerList[]	玩家列表	5	
commonPoker	PokerInfo[]	公共牌	6	
backPoker	PokerInfo[]	自己的底牌 2张	7	
smallBlindBet	int64	小盲注下注数量	8	
fullBlindBet	int64	全盲注下注数量	9	
waitDeclareTime	int32	表态等待时间(秒)	10	
curRewardPool	int64	当前底池	11	
addBetBaseValue	int64	加注时基本值	12	
curBet	int64	本轮底注	13	
playerBetInfo	PlayerBetInfo[]	玩家历史下注	14	
gameInfo	GameInfo	游戏信息	15	

请求表态

说明：玩家在自己表态倒数时间内可调用，用于告诉服务器自己的表态结果（跟，弃，加注等）

前端请求协议:packetc2l_req_declare

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_req_declare 10005
declareSel	int32	所选结果 EDeclare	2	
param	int32	参数，跟注时需用	3	

后端返回协议:packetl2c_req_declare_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_req_declare_result 15006
result	int32	结果	2	
curChip	int32	当前筹码	3	
hasBet	int32	已下注	4	

游戏开始通知

说明：服务端下发给客户端，在新一轮游戏开始的时候

后端返回协议:packetl2c_game_start_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_start_notify 15007
gameInfo	GameInfo	游戏信息	2	
playerList	PlayerInfo[]	玩家列表	3	
backPoker	PokerInfo[]	自己的底牌 2 张	4	
curRewardPool	int64	当前底池	5	

通知游戏等待（暂时不用）

后端返回协议:packetl2c_game_wait_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_wait_notify 15011

游戏结束通知，此时客户端结算

说明：服务端推送给客户端，一局游戏结束时发送，客户端收到消息后开始做相关结算

后端返回协议:packetl2c_game_end_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_end_notify 15009
curRewardPool	int64	当前底池	2	
playerList	BalancePlayerInfo[]	玩家列表	3	
isAllAbandon	bool	是否全部弃牌，除了一个人以外。这种情况下，由于公共牌没有发完，弃牌玩家没有牌型	4	
commonPoker	PokerInfo[]	公共牌信息	5	
playerBetInfo	PlayerBetInfo[]	玩家历史下注	6	

处于观战的玩家请求坐下

说明：没有坐在座位上的玩家可调用，只要在房间内任何阶段都可调用，无论游戏是否开始,有空座位则坐下成功

前端请求协议:packetc2l_req_sitdown

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_req_sitdown 10006

后端返回协议:packetl2c_req_sitdown_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_req_sitdown_result 15008
result	int32	结果	2	1.成功 非1.失败

错误码

```
enum e_msg_result_def
{
  e_rmt_unknow = 0; //未知错误
  e_rmt_success = 1; //成功
  e_rmt_fail = 2; //失败
  //其他
}
```

####

请求设置自动补充

说明：玩家用来设置每局结束后是否触发自动补充金币

现在每个房间都有配置进入的携带金币上限，设置自动补充后，每局结束后会自动补充金币

前端请求协议:packetc2l_req_supply

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_req_supply 10010
auto	bool	是否开启自动补充	2	

后端返回协议:packetl2c_req_supply_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_req_supply_result 15015
auto	bool	是否开启自动补充	2	
result	int32	结果	3	1.成功 非1.失败

错误码

```
enum e_msg_result_def
{
  e_rmt_unknow = 0; //未知错误
  e_rmt_success = 1; //成功
  e_rmt_fail = 2; //失败
  //其他
}
```

####

请求设置筹码(自动补充功能)

说明：设置自动补充到的具体指，下局开始时如果玩家携带金币不足则补充到该值

前端请求协议:packetc2l_req_chip

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_req_chip 10011
chip	int32	请求筹码	2	

后端返回协议:packetl2c_req_chip_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_req_chip_result 15016
result	int32	结果	2	1.成功 非1.失败

错误码

```
enum e_msg_result_def
{
  e_rmt_unknow = 0; //未知错误
  e_rmt_success = 1; //成功
  e_rmt_fail = 2; //失败
  //其他
}
```

####

购买筹码通知

说明：触发自动补充后，通知玩家（是广播消息，通知房间内所有人）

后端返回协议:packetl2c_buy_chip_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_buy_chip_notify 15010
playerList	PlayerInfo[]	玩家列表	2	

设置结束时是否亮牌(暂时不用)

前端请求协议:packetc2l_show_poker

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_show_poker 10007
isShow	bool	是否亮牌	2	

后端返回协议:packetl2c_show_poker_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_show_poker_result 15012
result	int32	结果	2	1.成功 非1.失败

玩家请求站起

说明：座位上的玩家如果不在游戏中，可主动站起，成功后进入旁观状态

前端请求协议:packetc2l_req_standup

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_req_standup 10008

后端返回协议:packetl2c_req_standup_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_req_standup_result 15013
result	int32	结果	2	1.成功 非1.失败

错误码

```
enum e_msg_result_def
{
  e_rmt_unknow = 0; //未知错误
  e_rmt_success = 1; //成功
  e_rmt_fail = 2; //失败
  //其他
}
```

####

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_get_room_info = 10001;//得到房间
    e_mst_c2l_join_table = 10002;//进入桌子()
    e_mst_c2l_leave_table = 10003;//离开桌子()

    // 请求桌子内场景信息
    e_mst_c2l_get_table_scene_info = 10004;

    // 请求表态
    e_mst_c2l_req_declare = 10005;

    // 处于观战的玩家请求坐下
    e_mst_c2l_req_sitdown = 10006;

    // 设置结束时是否亮牌
    e_mst_c2l_show_poker =  10007;

    // 坐下的玩家请求站起
    e_mst_c2l_req_standup = 10008;

    // 请求游戏状态
    e_mst_c2l_check_state = 10009;

    // 请求设置自动补充
    e_mst_c2l_req_supply = 10010;

    // 请求设置筹码
    e_mst_c2l_req_chip = 10011;

    // 服务端到客户端--------------------------------------------------------------------------
---

    e_mst_start_l2c = 15000;
    e_mst_l2c_get_room_info_result = 15001;//得到房间返回
    e_mst_l2c_join_table_result = 15002;//进入桌子
    e_mst_l2c_leave_table_result = 15003;//离开桌子

    // 获得桌子内场景信息
    e_mst_get_table_scene_info_result = 15004;

    // 游戏事件通知
    e_mst_l2c_game_event_notify = 15005;

    // 请求表态结果
    e_mst_l2c_req_declare_result = 15006;

    // 游戏开始通知
    e_mst_l2c_game_start_notify = 15007;

    // 处于观战的玩家请求坐下结果
```

```
    e_mst_l2c_req_sitdown_result = 15008;

    // 游戏结束通知
    e_mst_l2c_game_end_notify = 15009;

    // 购买筹码通知
    e_mst_l2c_buy_chip_notify = 15010;

    // 通知游戏等待
    e_mst_l2c_game_wait_notify = 15011;

    // 设置结束时是否亮牌结果
    e_mst_l2c_show_poker_result = 15012;

    // 坐下的玩家请求站起
    e_mst_l2c_req_standup_result = 15013;

    // 请求游戏状态返回
    e_mst_l2c_check_state_result = 15014;

    // 请求设置自动补充返回
    e_mst_l2c_req_supply_result = 15015;

    // 请求设置筹码返回
    e_mst_l2c_req_chip_result = 15016;

    e_mst_clend_index = 20000;
}
```