

- 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
    - [SrmjRoomConfig.xls](#)[房间表]
    - [SrmjRoomStockConfig.xls](#)[房间库存表]
  - 游戏通用结构
    - 通知类型
    - 卡数据
    - 结构卡数据
    - 换三张信息
    - 玩家信息
    - [room](#)场景信息
    - [playing](#)场景信息
    - 输家详情
    - 赢家详情
  - 协议列表
    - 进入房间
    - 请求场景信息
    - 准备
    - 玩家数据
    - 游戏开始
    - 出牌数据包
    - 操作数据包
    - 发送扑克
    - 游戏结束
    - 玩家通知消息
    - 换三张
    - 定缺
  - [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\_srmj\_protocol.proto  
game\_srmj\_def.proto

## 相关配置

### SrmjRoomConfig.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	

### SrmjRoomStockConfig.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	

字段名	描述	备注
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_listen = 4;                 // 听
    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
}
```

```
enum e_action_code {
    e_ac_null = 0x00;               // 没有类型
    // e_ac_left = 0x01;             // 左吃类型
    // e_ac_center = 0x02;           // 中吃类型
    // e_ac_right = 0x04;            // 右吃类型
    e_ac_peng = 0x08;               // 碰牌类型
    e_ac_gang = 0x10;               // 杠牌类型
    // e_ac_listen = 0x20;           // 听牌类型
    e_ac_hu = 0x40;                 // 胡类型
    // e_ac_pass_hu = 0x80;           // 过胡类型
    e_ac_qian_gang = 0x81;          // 抢杠类型
    e_ac_wind = 0x41;               // 刮风
    e_ac_add_gang = 0x42;           // 补杠
    e_ac_rain = 0x43;               // 下雨
    e_ac_tax = 0x44;                // 退税
    e_ac_scream = 0x45;             // 查大叫
    e_ac_pig = 0x46;                // 查花猪
}
```

### 卡数据

参数名	类型	描述	参数ID	默认值
card_id	int32	卡id	1	
card_val	int32	对家看着是背卡无此值	2	

参数名	类型	描述	参数ID	默认值
card	repeated <a href="#">card_data</a>	[1-13]扑克列表	1	

## 结构卡数据

参数名	类型	描述	参数ID	默认值
action_code	int32	<a href="#">e_action_code</a>	1	
cards	repeated int32	[3-4]扑克列表	2	
ming_gang	bool	[3-4]明杠	3	
jia_gang	bool	[3-4]加杠	4	
provider	int32	提供者	5	

参数名	类型	描述	参数ID	默认值
card	repeated <a href="#">pile_card_data</a>	[0-4]结构卡列表	1	

## 换三张信息

参数名	类型	描述	参数ID	默认值
old_card_id	int32	老卡id	1	
new_card_id	int32	新卡id	2	
new_card_val	int32	新卡val	3	

## 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	

参数名	类型	描述	参数ID	默认值
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	

playing场景信息

参数名	类型	描述	参数ID	默认值
banker_chair_id	int32	庄玩家	1	
current_user	int32	当前玩家	2	
out_card	int32	当前出卡[抢卡时有效]	3	
action_code	int32	e_action_code[抢卡时有效]	4	
remain_seconds	int32	剩余秒数	5	
user_trustee	repeated bool	[4]托管信息	6	
hu_cards	repeated cards_data	[4]已胡牌	7	
hand_cards	repeated cards_data	[4]手牌	8	
river_cards	repeated cards_data	[4]牌河	9	
pile_cards	repeated pile_cards_data	[4]结构扑克	10	
pass_times	repeated int32	[4]过胡倍数	11	
exchanged	repeated bool	[4]已换三张	12	
vacancy	repeated int32	[4]定缺 0:万子,1:索子,2:同子,x:未定	13	
dice_count	int32	骰子点数	14	
hu_list	repeated win_result	[4]已胡流水	15	
used_card_cnt	int32	已摸牌数	16	

## 输家详情

参数名	类型	描述	参数ID	默认值
lose_idx	int32	输家idx	1	
lose_account	int32	输金币数	2	
lose_limited	bool	输钱限定	3	

## 赢家详情

参数名	类型	描述	参数ID	默认值
win_idx	int32	赢家idx	1	
win_account	int32	赢金币数	2	
win_limited	bool	获利限定	3	
lose	repeated lose_result	输家详情	4	
fan_cnt	int32	总番数	5	
result1	repeated uint32	番型位	6	
result2	repeated uint32	番型位	7	
result3	repeated uint32	番型位	8	
win_root_cnt	int32	赢家根数	9	
user_action	int32	e_ac_hu, e_ac_wind, e_ac_add_gang, e_ac_rain, e_ac_tax, e_ac_scream, e_ac_pig	10	

## 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	
scene_info1	msg_room_scene_info	room场景信息	3	
scene_info3	msg_playing_scene_info	playing场景信息	4	
info_type	int32	哪种场景信息	5	
scene_info4	packetl2c_game_end	结算信息	6	

# 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	[4] 玩家信息	2	

# 游戏开始

说明:此后转入游戏状态

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
dice_count	int32	骰子点数	2	
banker_user	int32	庄家用户	3	
cards	repeated cards_data	[4]4家扑克列表	5	

# 出牌数据包



说明:游戏中玩家得出牌请求及通知  
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
card_id	int32	卡id	2	

后端返回协议:packetl2c\_out\_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_out_card_notify = 15014
action_user	int32	当前用户	2	
card	card_data	卡数据	3	

## 操作数据包

说明:游戏中玩家的吃碰听胡等操作  
后端返回协议:packetl2c\_operate\_notify,此消息指示玩家能对当前出的牌进行操作

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_operate_notify = 15015
user_action	int32	用户动作 e_action_code [x(要做动作), 0(等待)]	2	
card_val	int32	扑克数据	3	

前端请求协议:packetc2l\_operate\_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_operate_card = 10014
user_action	int32	用户动作 e_action_code	2	

参数名	类型	描述	参数ID	默认值
card_id	int32	卡id(主动抢卡有效)	3	

后端返回协议:packetl2c\_operate\_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_operate_result = 15019
action_user	int32	当前用户	2	
user_action	int32	用户动作 e_action_code	3	
provide_user	int32	供应用户	4	
card_val	int32	卡值	5	
pile_cards	repeated card_data	[3-4]结构扑克， 如card_id不为0， 要从手卡中删除	6	
hu_data	win_result	胡数据	7	

后端返回协议:packetl2c\_add\_win\_list,添加已胡流水

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_add_win_list = 15027
hu_data	win_result	胡数据	2	

## 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
current_user	int32	当前玩家	2	

参数名	类型	描述	参数ID	默认值
card	card_data	扑克数据	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
game_score	repeated int32	[4]当前金币	3	
win_account	repeated int32	[4]输赢金币	4	
win_fee	repeated int32	[4]赢家抽水	5	
hand_cards	repeated cards_data	[4]4家手卡	6	
pile_cards	repeated pile_cards_data	[4]4家结构卡	7	
hu_cards	repeated cards_data	[4]4家已胡牌	8	
hu_list	repeated win_result	已胡流水	9	

## 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	

# 换三张

前端请求协议:packetc2l\_exchange\_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_exchange_card = 10017
card	repeated int32	[3]id列表	2	

后端返回协议:packetl2c\_exchange\_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_exchange_card = 15025
exchange_type	int32	0:顺时针,1:逆时针,2:对家呼唤	2	
cards	exchange_card_data	[4]换牌数据	3	

# 定缺

前端请求协议:packetc2l\_decide\_vacancy

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_decide_vacancy = 10018
type	int32	0:万子,1:索子,2:同子	2	

后端返回协议:packetl2c\_decide\_vacancy

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_decide_vacancy = 15026

参数名	类型	描述	参数ID	默认值
types	repeated int32	[4]家定缺 0x00: 万子,0x10:索子,0x20: 同子	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_bu_hua = 10012;  
    e_mst_c2l_out_card = 10013;  
    e_mst_c2l_operate_card = 10014;  
    e_mst_c2l_gm_list = 10015;  
    e_mst_c2l_gm_kill = 10016;  
    e_mst_c2l_exchange_card = 10017;  
    e_mst_c2l_decide_vacancy = 10018;
```

```
    // 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_game_start = 15013;  
    e_mst_l2c_out_card_notify = 15014;  
    e_mst_l2c_operate_notify = 15015;  
    e_mst_l2c_game_end = 15016;  
    e_mst_l2c_send_all_playerinfos = 15017;  
    e_mst_l2c_game_notify = 15018;  
    e_mst_l2c_operate_result = 15019;  
    e_mst_l2c_send_card = 15020;  
    e_mst_l2c_bu_hua_notify = 15021;  
    e_mst_l2c_pass_hu_notify = 15022;  
    e_mst_l2c_other_player_cards = 15023;  
    e_mst_l2c_gm_list = 15024;  
    e_mst_l2c_exchange_card = 15025;  
    e_mst_l2c_decide_vacancy = 15026;  
    e_mst_l2c_add_win_list = 15027;
```

```
    e_mst_clend_index = 20000;
```

```
}
```