

- 奔驰宝马
 - 名词解释(或游戏文档)
 - 协议文件名
 - 游戏通用结构
 - 盘口ID
 - 游戏状态
 - 房间信息
 - 玩家信息
 - 玩家下注
 - 下注信息
 - 牌路说明
 - 玩家收益信息
 - 玩家下注信息,通知给GM
 - 协议列表
 - 进入房间
 - 相关配置
 - 获取房间信息
 - 检测游戏状态
 - 请求场景信息
 - 下注
 - 相关配置
 - 续压
 - 相关配置
 - 清空下注
 - 相关配置
 - 申请上庄
 - 相关配置
 - 申请下庄
 - 抢庄
 - 相关配置
 - 请求玩家列表
 - 请求上庄列表
 - 请求牌路
 - 通知开始押注
 - 通知开奖
 - 相关配置
 - 通知所有玩家下注信息
 - 通知抢庄信息
 - 通知更换庄家
 - 游戏控制
 - 库存信息
 - 通知所有玩家下注信息
 - 通知庄家信息
 - 协议ID枚举

奔驰宝马

名词解释(或游戏文档)

暂无

协议文件名

game_benzbmw_protocol.proto
game_benzbmw_def.proto
msg_type_def.proto

游戏通用结构

盘口ID

- 0 银标大众
- 1 银标奔驰
- 2 银标宝马
- 3 银标保时捷
- 4 金标大众
- 5 金标奔驰
- 6 金标宝马
- 7 金标保时捷

游戏状态

```
enum e_game_state {  
    e_state_game_begin = 0; //游戏开始  
    e_state_game_bet = 1;    //押注期间  
    e_state_game_award = 2;  //开奖期间  
}
```

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

参数名	类型	描述	参数ID	默认值
player_id	int32	玩家ID	1	
player_nickname	string	玩家昵称	2	
player_head_frame	int32	头像框	3	
player_head_custom	string	玩家头像	4	
player_gold	int64	玩家金币	5	
player_sex	int32	玩家性别	6	
player_vip_lv	int32	Vip等级	7	
win_count	int32	获胜数	8	
bets	int32	下注数	9	
play_cnt	int32	局数	10	
other_bets	repeated int32	每个盘口下注	11	
other_win	repeated int32	每个盘口输赢金币	12	

玩家下注

参数名	类型	描述	参数ID	默认值
player_id	int32	玩家ID	1	
player_bets	int64	下注	2	

下注信息

参数名	类型	描述	参数ID	默认值
bet_golds	int64	下注	1	
max_bet_count	int64	最大下注	2	
master_bets	msg_master_bets	6个机器人的下注信息	3	

牌路说明

牌路列表是个整型数组，每个元素代表一局开奖结果.具体对应关系，见[盘口ID](#).

玩家收益信息

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_baseInfo	player_info	玩家信息	1	
betCount	int32	下注金额	2	
income	int32	返奖	3	

玩家下注信息,通知给GM

参数名	类型	描述	参数ID	默认值
player_info	player_info	玩家信息	1	
bet_info_list	int32数组	下注列表	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 = 10002
roomid	int32	房间ID	2	

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 = 15002
result	msg_type_def.e_msg_result_def	结果状态	2	

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

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
    e_rmt_room_full = 12; // 房间已满
}
```

相关配置

表名	字段名	描述	备注
BenzBmwRoomConfig	PlayerMaxCount	房间最大人数	包括机器人

获取房间信息

说明:获取房间列表
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	

检测游戏状态

说明:重连之后判断，若在桌子中，进入断线重连流程
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 = 10008

后端返回协议:packetl2c_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 = 15013
is_intable	bool	是否在桌子中	2	

请求场景信息

说明:进入房间和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_room_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_room_scene_info = 10007

后端返回协议:packetl2c_get_room_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_get_room_scene_info = 15010
room_id	int32	房间Id	2	
room_state	e_game_state	房间状态	3	
cd_time	int32	当前房间状态倒计时	4	
bet_info_list	repeated msg_betinfo	盘口下注列表	5	
history_list	int32数组	牌路列表	6	
is_can_rob_banker	bool	是否可以抢庄	7	
banker_info	player_info	庄家信息	8	
continue_count	int32	庄家坐庄次数	9	
self_bet_list	int32数组	自己的下注列表	10	

下注

说明:收到开始下注通知后，玩家可以开始下注
前端逻辑判断

- 1.自己是庄家不能下注.
- 2.判断当前金币是否大于下注金额.
- 3.判断当前盘口限红.限红指的是每个玩家在每个盘口的最大下注金额.
- 4.本局开始时,身上携带的金币,高于下注限制才能下注.
- 5.盘口ID在范围内.

前端请求协议:packetc2l_add_bet

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_add_bet = 10004
bet_index	int32	盘口ID	2	

参数名	类型	描述	参数ID	默认值
bet_count	int32	下注金额	3	

后端返回协议:packetl2c_add_bet_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_add_bet = 15004
result	msg_type_def.e_msg_result_def	结果状态	2	
bet_index	int32	盘口ID	3	
bet_count	int32	下注金额	4	

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
    e_rmt_gold_not_enough = 10; // 金币不足
    e_rmt_bet_index_error = 21; //押注序号不对
    e_rmt_outof_bet_limit = 22; //超过房间押注上限
    e_rmt_no_find_table = 23; //没有找到桌子
    e_rmt_banker_not_bet = 45; //庄家不能押注
    e_rmt_other_betgold_is_full = 58; //超过限红
    e_rmt_betgold_not_enough = 92; // 本局开始时拥有的金币，低于下注限制
}
```

相关配置

表名	字段名	描述	备注
BenzBmwBaseConfig	BetTime	下注时间	下注阶段持续时间
BenzBmwRoomConfig	BetLimit	限红	每个玩家每个盘口最大下注金额
BenzBmwRoomConfig	BetCondition	下注限制	本局开始时携带的金币,大于下注限制,才能下注

续压

说明:重复上一局的下注，若当前已下注，不可续压

前端逻辑判断

- 1.自己是庄家不能续压.
- 2.已下注不能续压.
- 3.本局开始时,身上携带的金币,高于下注限制才能下注.

前端请求协议:packetc2l_repeat_bet

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_repeat_bet = 10005

后端返回协议:packetl2c_repeat_bet_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_repeat_bet = 15005
result	msg_type_def.e_msg_result_def	结果状态	2	

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
    e_rmt_gold_not_enough = 10; // 金币不足
    e_rmt_outof_bet_limit = 22; //超过房间押注上限
    e_rmt_no_find_table = 23; //没有找到桌子
    e_rmt_banker_not_bet = 45; //庄家不能押注
    e_rmt_other_betgold_is_full = 58; //超过限红
    e_rmt_can_not_bet_hasbet = 85; //玩家下注了不能续压
    e_rmt_betgold_not_enough = 92; // 本局开始时拥有的金币，低于下注限制
}
```

相关配置

表名	字段名	描述	备注
BenzBmwBaseConfig	BetTime	下注时间	下注阶段持续时间
BenzBmwRoomConfig	BetCondition	下注限制	本局开始时携带的金币,大于下注限制,才能下注

清空下注

说明:清空下注,房间状态为押注状态下可调用
前端逻辑判断

1.下注阶段才能清空下注.

前端请求协议:packetc2l_clear_bet

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_clear_bet = 10006

后端返回协议:packetl2c_clear_bet_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_clear_bet = 15006
result	msg_type_def.e_msg_result_def	结果状态	2	

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
    e_rmt_no_can_bet = 20; //押注时间已过
}
```

相关配置

表名	字段名	描述	备注
BenzBmwRoomConfig	BetTime	下注时间	下注阶段持续时间

申请上庄

说明:申请加入上庄列表
前端请求协议:packetc2l_ask_for_banker

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_ask_for_banker = 10010

后端返回协议:packetl2c_ask_for_banker_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_ask_for_banker = 15016
result	msg_type_def.e_msg_result_def	结果状态	2	

前端逻辑判断
1.携带的金币是否高于上庄金额

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
    e_rmt_gold_not_enough = 10; //金币低于上庄金额
    e_rmt_banker_is_full = 46; //上庄列表已满
    e_rmt_has_in_banker_list = 52; //已经在上庄列表中
    e_rmt_now_is_banker = 53; //现在是庄家
}
```

相关配置

表名	字段名	描述	备注
BenzBmwRoomConfig	BankerCondition	上庄金额	

申请下庄

说明:申请下庄
前端请求协议:**packetc2l_leave_banker**

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_banker = 10011

后端返回协议:**packetl2c_leave_banker_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_banker = 15017
result	msg_type_def.e_msg_result_def	结果状态	2	

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
}
```

抢庄

说明:花钱抢庄,若当前庄家不在上庄保护期
前端请求协议:packetc2l_ask_for_first_banker

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_ask_first_for_banker = 10012

后端返回协议:packetl2c_ask_for_first_banker_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_ask_first_for_banker = 15018
result	msg_type_def.e_msg_result_def	结果状态	2	

前端逻辑判断

1.携带的金币是否高于上庄金额

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
    e_rmt_gold_not_enough = 10; // 金币不足
    e_rmt_now_is_banker = 53; //现在是庄家
    e_rmt_now_is_you = 63; // 庄家已经是自己了
}
```

相关配置

表名	字段名	描述	备注
BenzBmwBaseConfig	PminBankerCount	上庄保护局数	上庄保护期间不能被抢庄
BenzBmwRoomConfig	FirstBankerCost	抢庄花费	
BenzBmwRoomConfig	AddBankerCost	每次抢庄竞价累计值	本局每次抢庄,价格提升

请求玩家列表

说明:请求当前房间的玩家列表
前端请求协议:packetc2l_ask_for_player_list

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_ask_player_list = 10013

后端返回协议:packetl2c_ask_for_player_list_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_ask_player_list = 15020
player_list	player_info数组	玩家列表	2	

请求上庄列表

说明:请求当前房间的上庄列表

前端请求协议:packetc2l_ask_for_banker_list

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_ask_banker_list = 10014

后端返回协议:packetl2c_ask_for_banker_list_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_ask_banker_list = 15021
banker_list	player_info数组	上庄列表	2	

请求牌路

说明:请求当前房间的牌路

前端请求协议:packetc2l_ask_for_history_list

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_ask_history_list = 10015

后端返回协议:packetl2c_ask_for_history_list_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_ask_history_list = 15022
history_list	int32数组	牌路列表	2	

通知开始押注

说明:后端主动通知前端
后端通知协议:packetl2c_bc_begin_bet

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_bc_begin_bet = 15007
is_can_rob_banker	bool	是否可以抢庄	2	

通知开奖

说明:后端主动通知前端
后端通知协议:packetl2c_bc_begin_award

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_bc_begin_award = 15008
cd_time	int32	开奖阶段倒计时， 这个时间给前端播放结算动画	2	
result_index	int32	开奖结果	3	
banker_result	int32	庄家盈利	4	
playerIncomeList	PlayerIncomeInfo数组	玩家收益列表	5	

相关配置

表名	字段名	描述	备注
BenzBmwBaseConfig	BroadcastGold	广播金币	赢钱超过该金额,发广播

通知所有玩家下注信息

说明:通知所有玩家下注信息
后端通知协议:packetl2c_bc_total_bet_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_l2c_bc_total_bet_info = 15009
bets	repeated msg_betinfo	每个盘口的下注信息	2	

通知抢庄信息

说明:通知抢庄信息
后端通知协议:packetl2c_bc_rob_banker_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_l2c_bc_rob_banker_info = 15023
player_id	int32	玩家ID	2	
pay_count	int32	抢庄花费	3	

通知更换庄家

说明:通知更换庄家
后端通知协议:packetl2c_bc_change_banker

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_bc_change_banker = 15019
banker_info	player_info	庄家信息	2	
is_rob	bool	是否抢庄	3	
old_banker_id	int32	旧庄家ID	4	

游戏控制

说明:控制客户端+超级账号，可以控制游戏结果
前端逻辑判断
1.当前账号是超级账号，并且有控制模块，才显示控制界面

前端请求协议:packetc2l_control_change_result

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id		
command	int32	控制开奖结果,盘口ID+1		

后端返回协议:packetl2c_control_change_result

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
packet_id	e_server_msg_type	协议Id		
result	msg_type_def.e_msg_result_def	结果状态		

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
}
```

库存信息

说明:超级账号才会收到这条协议,由后端主动推送

后端协议:packetl2c_bc_debuginfo

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id		
player_id	int32	玩家ID		
debug_infos	string	库存信息 lua对象的字符串		
debug_infos.TotalProfit	string	总抽水		
debug_infos.TotalStock	string	总库存		

通知所有玩家下注信息

说明:每个玩家的具体下注信息,超级账号才会收到这条协议,由后端主动推送

后端协议:packetl2c_bc_every_bet_info

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id		
user_info_list	user_meg_bet_info数组	玩家下注列表		

通知庄家信息

说明:庄家信息,类型。超级账号才会收到这条协议,由后端主动推送

后端协议:packetl2c_notice_bank_info

参数名	类型	描述	参数ID	默认值
packet_id	e_server_msg_type	协议Id		
player_info	player_info	庄家信息		
type	int32	庄家类型 0系统小庄 1 机器人 2 玩家		

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_enter_room = 10002; //进入房间
    e_mst_c2l_leave_room = 10003; //离开房间
    e_mst_c2l_add_bet = 10004; //下注
    e_mst_c2l_repeat_bet = 10005; //重复押注
    e_mst_c2l_clear_bet = 10006; //押注清零
    e_mst_c2l_get_room_scene_info = 10007; //获得桌子内信息
    e_mst_c2l_check_state = 10008; //检测游戏状态
    e_mst_c2l_gm = 10009; //GM
    e_mst_c2l_ask_for_banker = 10010; //上庄
    e_mst_c2l_leave_banker = 10011; //下庄
    e_mst_c2l_ask_first_for_banker = 10012; //抢庄
    e_mst_c2l_ask_player_list = 10013; //请求玩家列表
    e_mst_c2l_ask_banker_list = 10014; //请求上庄列表
    e_mst_c2l_ask_history_list = 10015; //请求牌路

    e_mst_c2l_control_change_result = 10016; //控制客户端提示服务器是否放分

    // 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_enter_room_result = 15002; //进入房间
    e_mst_l2c_leave_room_result = 15003; //离开房间
    e_mst_l2c_add_bet = 15004; //下注
    e_mst_l2c_repeat_bet = 15005; //重复押注
    e_mst_l2c_clear_bet = 15006; //押注清零
    e_mst_l2c_bc_begin_bet = 15007; //通知开始下注
    e_mst_l2c_bc_begin_award = 15008; //通知开奖
    e_mst_l2c_bc_total_bet_info = 15009; //通知所以玩家下注信息
    e_mst_l2c_get_room_scene_info = 15010; //获得桌子内信息
    e_mst_l2c_check_state_result = 15013; //检测游戏状态
    e_mst_l2c_bc_accept_gift = 15015; //广播桌子内玩家收到礼物
    e_mst_l2c_ask_for_banker = 15016; //上庄
    e_mst_l2c_leave_banker = 15017; //下庄
    e_mst_l2c_ask_first_for_banker = 15018; //抢庄
    e_mst_l2c_bc_change_banker = 15019; //通知更换庄家
    e_mst_l2c_ask_player_list = 15020; //请求玩家列表返回
    e_mst_l2c_ask_banker_list = 15021; //请求上庄列表返回
    e_mst_l2c_ask_history_list = 15022; //请求牌路返回
    e_mst_l2c_bc_rob_banker_info = 15023; //通知抢庄信息

    e_mst_l2c_bc_every_bet_info = 15500; //每个玩家的下注信息
    e_mst_l2c_control_change_result = 15502; //

    e_mst_l2c_bc_debuginfo = 15505; //库存等信息
    e_mst_l2c_notice_gm_bank_info = 15506; //通知每个控制端的当前庄家信息

    e_mst_clend_index = 20000;
}

```