

- 捕鱼(四款捕鱼产品通用)
 - 名词解释(或游戏文档)
 - 协议文件名
 - 游戏通用结构
 - 鱼ID
 - Buff列表
 - 相关配置
 - 物品信息
 - 相关配置
 - 座位信息
 - 鱼初始化信息
 - 相关配置
 - 鱼阵信息
 - 相关配置
 - 场景信息
 - 鱼信息
 - 协议列表
 - 检测游戏状态
 - 开启/关闭功能
 - 请求场景信息
 - 改变炮倍
 - 相关配置
 - 主动请求同步金币
 - 广播有玩家进入
 - 广播有玩家离开
 - 机器人进入
 - 机器人离开
 - 广播玩家倍率修改
 - 广播玩家金币变更(任务,充值等)
 - 新刷出了一条鱼
 - 新刷出了一个鱼阵
 - 请求开炮
 - 广播玩家开炮
 - 玩家开炮,击中鱼
 - 玩家开炮,击中多条鱼(暂时没用到,保留)
 - 鱼死亡通知
 - 大群鱼死亡
 - 变更场景,刷水
 - 增加BUFF
 - 移除BUFF
 - 改变炮台

- [锁定鱼](#)
- [广播锁定鱼](#)
- [广播冻结](#)
- [切换场景](#)
- [场景倒计时\(废弃\)](#)
- [通知击中鱼\(废弃\)](#)
- [通知玩家的最大能量值发生变化\(废弃\)](#)
- [库存信息](#)
- [协议ID枚举](#)

捕鱼(四款捕鱼产品通用)

名词解释(或游戏文档)

暂无

协议文件名

fish_def.proto

fish_logic.proto

fish_protocol.proto

msg_type_def.proto

游戏通用结构

鱼ID

详情见各个捕鱼游戏的配置文件FishFishConfig.xls

- 1 蜗牛鱼
- 2 小绿鱼
- 3 石斑鱼
- 4 大眼鱼
- 5 阿扁鱼
- 6 小丑鱼
- 7 河豚鱼
- 8 蓝鱼
- 9 灯笼鱼
- 10 神仙鱼

- 11 海龟
- 12 蝴蝶鱼
- 13 孔雀鱼
- 14 剑鱼
- 15 蝙蝠鱼
- 19 小海豚
- 20 黄金鲨
- 21 金龙
- 22 绿蟾
- 23 火龙珠
- 24 定身金龟
- 25 深水炸弹
- 26 三连坐
- 27 四连坐
- 28 五连坐
- 29 红鱼 蜗牛鱼
- 30 红鱼 小绿鱼
- 31 红鱼 石斑鱼
- 32 红鱼 大眼鱼
- 33 红鱼 阿扁鱼
- 34 红鱼 小丑鱼
- 35 红鱼x2 蜗牛鱼
- 36 红鱼x2 小绿鱼
- 37 红鱼x2 石斑鱼
- 38 红鱼x2 大眼鱼
- 39 红鱼x2 阿扁鱼
- 40 红鱼x2 小丑鱼
- 41 BOSS金蟾

Buff列表

参数名	类型	描述
buffid	int32	BuffID
outtime	int32	结束时间

相关配置

表名	字段名	描述	备注
FishBuffConfig	ID	BuffID	每个BUFF的ID，目前只有锁定，急速，散弹，自动，其他都是废弃的

物品信息

参数名	类型	描述
item_id	int32	物品ID
item_count	int32	数量
end_time	int32	到期时间

相关配置

表名	字段名	描述	备注
FishItemConfig	ItemID	物品ID	基本都是废弃的，只有锁定,急速,自动还有用

座位信息

参数名	类型	描述
seat_index	int32	座位号
player_id	int32	玩家ID
nickname	string	玩家昵称
turret_rate	int32	炮倍
turret_id	int32	炮台ID
bufflist	msg_buff_info	Buff列表
turret_power	int32	当前炮台能量(废弃)
max_turret_power	int32	最大炮台能量(废弃)
photo_frame	int32	头像框
icon_custom	string	头像
sex	int32	性别
vip	int32	VIP等级
player_ticket	int32	点券(废弃)

鱼初始化信息

参数名	类型	描述
id	int32	鱼的序列ID
fish_id	int32	鱼的配置ID
life_time	int32	鱼存活了多久,单位毫秒
route_param	int32数组	鱼游路径参数

相关配置

表名	字段名	描述	备注
FishNormalOutFishConfig	ID	ParamID	鱼的配置ID或鱼阵ID
FishNormalOutFishConfig	OutWeight	出鱼权值	根据权重决定刷出什么鱼和鱼阵
FishNormalOutFishConfig	RouteGroup	鱼游路径分组	关联路径分组配置
FishNormalOutFishConfig	GroupID	鱼分组	
FishNormalOutFishConfig	GroupMaxFish	分组最大鱼数	每个分组,同时最多存在多少条鱼
FishRouteGroupConfig	RouteIDs	鱼游路径分组	鱼会在这个路径分组中,随机选一条鱼游路径
RouteDataCFG.xml	ID	路径ID	各种鱼游路径

鱼阵信息

参数名	类型	描述
generator_id	int32	鱼阵ID
life_time	int32	鱼阵存活了多久,单位毫秒
start_id	int32	鱼的开始序号
route_id	int32	路径ID,对应RouteDataCFG.xml配置文件中的ID
pos_x	int32	鱼阵位置

参数名	类型	描述
pos_y	int32	鱼阵位置

相关配置

表名	字段名	描述	备注
FishGeneratorConfig	ID	鱼阵ID	
FishGeneratorConfig	Type	鱼阵类型	
FishGeneratorConfig	Param	鱼阵参数	根据不同的鱼阵类型,参数意义不同,详情看配置文件

场景信息

参数名	类型	描述
sceneid	int32	场景ID
roomid	int32	房间ID
seat_info	msg_seat_info	座位信息
fish_infos	msg_fish_init 数组	鱼初始化信息
generator_infos	msg_generator_init 数组	鱼阵初始化信息

鱼信息

参数名	类型	描述
id	int32	鱼的序列ID
fish_id	int32	鱼的配置ID
route_id	int32	鱼的路径ID,对应RouteDataCFG.xml配置文件中的ID

协议列表

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
packet_id	e_server_msg_type	协议Id

后端返回协议:packetl2c_check_state_result

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
is_intable	bool	是否在桌子中
itemlist	msg_item	物品列表
turret_id	int32	炮台ID
gamelvl	int32	捕鱼等级(废弃)
gameexp	int32	捕鱼经验值(废弃)
getgift	int32	最后抽奖时间(废弃)

开启/关闭功能

说明:目前只支持锁定，自动，急速,其他废弃
前端请求协议:packetc2l_use_item

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
item_id	int32	功能ID， 1锁定,2急速,4自动

| pos_x| int32| 废弃 |
| pos_x| int32| 废弃 |

后端返回协议:packetl2c_use_item_result

参数名	类型	描述
packet_id	e_server_msg_type	协议Id

参数名	类型	描述
item_id	int32	功能ID， 1锁定,2急速,4自动
result	msg_type_def.e_msg_result_def	结果状态
item_count	int32	使用数量,通常一直都是1

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
}
```

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_scene_info

参数名	类型	描述
packet_id	e_server_msg_type	协议Id

后端返回协议:packetl2c_get_scene_info_result

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
scene_info	msg_scene_info	场景信息

改变炮倍

说明:修改当前炮倍,如果修改炮倍失败,才返回协议
前端逻辑判断
1.前端发送协议，不等待返回,如果修改炮倍失败才返回协议.
2.炮倍可选范围,先从配置文件中读取,禁止发可选范围外的炮倍.

前端请求协议:packetc2l_change_rate

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
bet_index	int32	盘口ID
bet_count	int32	下注金额

后端返回协议:packetl2c_change_rate_result

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
result	msg_type_def.e_msg_result_def	结果状态
old_rate	int32	原炮倍

错误码

```
enum e_msg_result_def
{
    e_rmt_fail = 2; //失败
}
```

相关配置

表名	字段名	描述	备注
FishRoomConfig	RateList	可选炮倍范围	

主动请求同步金币

说明:捕鱼每分钟同步一次数据到world.
如果金币没了,弹出领救济金,需要前端主动请求,把金币同步到world防止领救济金失败.

前端逻辑判断

1.钱不够，需要领救济金的时候，才发送协议.

前端请求协议:packetc2l_sync_gold

参数名	类型	描述
packet_id	e_server_msg_type	协议Id

广播有玩家进入

说明:有玩家进入桌子,根据座位信息, 把玩家显示出来
后端通知协议:packetl2c_bc_enter_seat

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
seat_info	msg_seat_info	座位信息

广播有玩家离开

说明:有玩家进入离开,根据玩家ID,把玩家移除显示
后端通知协议:packetl2c_bc_leave_seat

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
player_id	int32	玩家ID

机器人进入

说明:机器人进入桌子通知,不会发给前端,只在服务器内部处理
后端通知协议:packetl2c_robot_enter

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
seat_info	msg_seat_info	座位信息
scene_id	int32	场景ID

机器人离开

说明:机器人离开桌子通知,不会发给前端,只在服务器内部处理
后端通知协议:packetl2c_robot_leave

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
player_id	int32	玩家ID

广播玩家倍率修改

说明:玩家倍率修改成功后,广播通知桌子内的所有玩家
后端通知协议:packetl2c_bc_change_rate

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
player_id	int32	玩家ID
turret_rate	int32	玩家炮倍ID

广播玩家金币变更(任务,充值等)

说明:游戏外部的原因, 导致玩家金币发生变化,通知桌子内的其他玩家
后端通知协议:packetl2c_bc_change_attr

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
player_id	int32	玩家ID
change_gold	int64	变化的金币
change_vip	int32	变化的vip等级
change_ticket	int32	废弃

新刷出了一条鱼

说明:在鱼创建出来的时候通知给前端
后端通知协议:packetl2c_bc_create_fish

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
fish_info	msg_fish_info	鱼信息

新刷出了一个鱼阵

说明:在鱼阵创建出来的时候通知给前端
后端通知协议:packetl2c_bc_create_fish

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
generator_id	int32	鱼阵ID
start_id	int32	鱼的开始序号
route_id	int32	路径ID,对应RouteDataCFG.xml配置文件中的ID
pos_x	int32	鱼阵位置
pos_y	int32	鱼阵位置
is_flash	int32	是否需要播放刷水动画

请求开炮

说明:发射炮弹时,前端通知后端发射了炮弹,除非失败,否则不会返回协议
前端逻辑判断

1.如果收到返回协议,将炮弹消耗的金额加回去.

前端请求协议:packetc2l_manual_fire

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
bullet_id	int32	炮弹ID
bullet_angle	int32	炮弹角度
is_multi	bool	是否开启散弹

参数名	类型	描述
is_fast	bool	是否开启急速
bullet_type	int32	炮弹类型,固定1
bullet_rate	int32	炮弹倍率

后端返回协议:packetl2c_manual_fire_fail

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
bullet_cost	int32	炮弹消耗

广播玩家开炮

说明:玩家开炮成功,广播通知

前端逻辑判断

1.收到这条协议，根据player_id判断，显示其他玩家开炮的动画

前端请求协议:packetl2c_bc_manual_fire

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
bullet_id	int32	炮弹ID
player_id	int32	玩家ID
bullet_angle	int32	炮弹角度
is_multi	bool	是否开启散弹
bullet_type	int32	炮弹类型,固定1

玩家开炮,击中鱼

说明:玩家开炮击中鱼时发送,不会返回协议

前端请求协议:packetc2l_hit_fish

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
bullet_id	int32	炮弹ID
fish_id	int32	击中的鱼序列ID

玩家开炮,击中多条鱼(暂时没用到,保留)

说明:暂时没用到,保留
前端请求协议:packetc2l_hit_fishs

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
hit_type	int32	击中类型
bullet_id	int32	炮弹ID
fish_ids	int32数组	击中的鱼序列ID列表
hit_param	int32	击中参数

鱼死亡通知

说明:玩家开炮,击中鱼后,如果成功击杀鱼,广播通知
前端逻辑判断

1.给kill_id对应的玩家,加钱,播放飞金币动画

通知协议:packetl2c_bc_fish_die

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
fish_id	int32	鱼序列ID
kill_id	int32	击杀鱼的玩家
base_score	int32	返奖
add_score	int32	废弃

参数名	类型	描述
ticket	int32	废弃
getstar	int32	废弃

大群鱼死亡

说明:击杀红鱼,或者触发炸弹,可能会杀死多条鱼,广播通知
前端逻辑判断

1.给player_id对应的玩家,加钱,播放飞金币动画

通知协议:packetl2c_bc_fishs_die

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
player_id	int32	击杀鱼的玩家
die_type	int32	死亡类型,1红鱼,2全屏炸弹
die_params	int32数组	死亡参数,1.红鱼ID 位置X 位置Y 2,全屏炸弹ID 位置X 位置Y
total_score	int32	总返奖
base_rate	int32	炮倍
add_score	int32	废弃
ticket	int32	废弃
getstar	int32	废弃

变更场景,刷水

说明:每隔一分钟,触发一次,随机刷一个特定鱼阵
通知协议:packetl2c_bc_change_scene

参数名	类型	描述
packet_id	e_server_msg_type	协议Id

参数名	类型	描述
scene_id	int32	场景ID
large_fish_id	int32	特定鱼阵的ID

增加BUFF

说明:实际就是通知有玩家开启,急速,锁定,自动
通知协议:packetl2c_bc_change_scene

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
buff_id	int32	buffid 1锁定,2急速,4自动
player_id	int32	玩家ID
buff_time	int32	持续时间,-1无限

移除BUFF

说明:实际就是通知有玩家关闭,急速,锁定,自动
通知协议:packetl2c_bc_removebuff

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
buff_id	int32	buffid 1锁定,2急速,4自动
player_id	int32	玩家ID

改变炮台

说明:炮台变更,通知其他玩家
通知协议:packetl2c_bc_changeturret

参数名	类型	描述
-----	----	----

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
turret_id	int32	炮台ID
player_id	int32	玩家ID

锁定鱼

说明:前端请求，需要锁定某条鱼,不等待返回协议
前端请求协议:packetc2l_lock_fish

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
fish_id	int32	鱼序列ID

广播锁定鱼

说明:锁定鱼成功之后,广播通知
通知协议:packetl2c_bc_lock_fish

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
player_id	int32	玩家ID
fish_id	int32	鱼序列ID

广播冻结

说明:广播冻结,时间停止，全局鱼不动,击杀特殊鱼,会触发
通知协议:packetl2c_bc_freeze

参数名	类型	描述
packet_id	e_server_msg_type	协议Id

参数名	类型	描述
freeze_Time	int32	冻结时间
is_show	bool	是否播放特效

切换场景

说明:广播通知场景变化，主要是从鱼潮场景变回正常场景
通知协议:packetl2c_bc_change_scene_bg

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
scene_id	int32	场景ID,固定5

场景倒计时(废弃)

说明:海盗秘境场景结束倒计时
通知协议:packetl2c_bc_scene_time

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
scene_time	int32	场景结束倒计时

通知击中鱼(废弃)

说明:闪电链,冰冻炮弹,击中鱼的通知
通知协议:packetl2c_bc_hit_fishs

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
player_id	int32	玩家ID
hit_type	int32	击中类型

参数名	类型	描述
fish_ids	int32数组	击中的鱼列表
params	int32数组	参数列表

通知玩家的最大能量值发生变化(废弃)

说明:切换炮台,升级,导致能量最大值发生变化的通知

通知协议:packetl2c_bc_change_max_power

参数名	类型	描述
packet_id	e_server_msg_type	协议Id
player_id	int32	玩家ID
max_power	int32	最大能量值

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
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

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;        //获取游戏信息
    e_mst_c2l_enter_table = 10002;          //进入游戏
    e_mst_c2l_leave_table = 10003;          //离开游戏
    e_mst_c2l_check_state = 10004;          //检测状态
    e_mst_c2l_try_enter_table = 10005;       //尝试进入游戏

    e_mst_c2l_get_scene_info = 10100;       //获取场景信息
    e_mst_c2l_change_seat = 10101;          //请求改变座位
    e_mst_c2l_change_rate = 10102;          //请求改变倍率
    e_mst_c2l_manual_fire = 10103;          //手动发炮
    e_mst_c2l_hit_fish = 10104;              //击中鱼
    e_mst_c2l_buy_item = 10105;              //购买道具
    e_mst_c2l_use_item = 10106;              //使用道具
    e_mst_c2l_get_shoplist = 10107;          //获取商品列表
    e_mst_c2l_hit_fishes = 10108;           //击中鱼
    e_mst_c2l_lock_fish = 10109;            //锁定鱼
    e_mst_c2l_player_levelup = 10110;        //玩家升级
    e_mst_c2l_usetick_levelup = 10111;       //使用礼券升级
    e_mst_c2l_change_table = 10112;          //换桌子
    e_mst_c2l_try_turret = 10113;           //试用炮台结束
    e_mst_c2l_try_turret_state = 10114;      //请求试用状态
    e_mst_c2l_use_turret_power = 10115;      //请求使用大招
    e_mst_c2l_turret_rotate = 10116;         //炮台旋转
    e_mst_c2l_missile_fire = 10117;          //导弹发射
    e_mst_c2l_sync_gold = 10118;             //同步金额

    // 服务端到客户端-----

    e_mst_start_l2c = 15000;
    e_mst_l2c_get_room_info_result = 15001;   //返回游戏信息
    e_mst_l2c_enter_table_result = 15002;      //返回进入游戏
    e_mst_l2c_leave_table_result = 15003;      //离开游戏
    e_mst_l2c_check_state_result = 15004;      //返回状态
    e_mst_l2c_try_enter_table_result = 15005;   //尝试进入游戏结果

    e_mst_l2c_get_scene_info_result = 15100;   //获取场景信息结果
    e_mst_l2c_change_seat_result = 15101;      //改变座位结果
    e_mst_l2c_player_levelup_result = 15102;    //返回玩家升级
    e_mst_l2c_usetick_levelup_result = 15103;   //返回礼券升级
    e_mst_l2c_change_table_result = 15104;      //切换桌子返回
    e_mst_l2c_try_turret_result = 15105;        //试用炮台
    e_mst_l2c_try_turret_state_result = 15106;  //试用状态
    e_mst_l2c_use_turret_power_result = 15107;  //使用大招结果
    e_mst_l2c_give_item_result = 15108;         //获得物品
    e_mst_l2c_player_auto_levelup = 15109;     //玩家升级
    e_mst_l2c_change_rate_result = 15110;      //改变倍率
    e_mst_l2c_manual_fire_fail = 15111;        //发炮失败

    //同步信息
    e_mst_l2c_bc_enter_seat = 15201;           //广播进入座位

```

```

e_mst_l2c_bc_leave_seat = 15202; //广播离开座位
e_mst_l2c_bc_change_rate = 15203; //广播改变倍率
e_mst_l2c_bc_create_fish = 15204; //广播创建鱼
e_mst_l2c_bc_manual_fire = 15205; //广播手动发炮
e_mst_l2c_bc_fish_die = 15206; //广播鱼死亡
e_mst_l2c_bc_change_attr = 15207; //广播玩家金币变更
e_mst_l2c_bc_create_generator = 15208; //广播鱼阵
e_mst_l2c_bc_change_scene = 15209; //广播刷水
e_mst_l2c_buy_item_result = 15210; //购买道具返回
e_mst_l2c_use_item_result = 15211; //使用道具返回
e_mst_l2c_get_shoplist_result = 15212; //获取商品列表返回
e_mst_l2c_bc_addbuff = 15213; //增加buff
e_mst_l2c_bc_removebuff = 15214; //移除buff
e_mst_l2c_bc_changeturret = 15215; //改变炮台
e_mst_l2c_bc_lock_fish = 15216; //锁定炮台
e_mst_l2c_bc_freeze = 15217; //全屏静止
e_mst_l2c_bc_fishs_die = 15218; //广播鱼死亡
e_mst_l2c_bc_specialfish_die = 15219; //广播火龙珠死亡
e_mst_l2c_bc_change_scene_bg = 15220;
e_mst_l2c_bc_scene_time = 15221;
e_mst_l2c_bc_hit_fishs = 15223; //击中鱼
e_mst_l2c_bc_turret_rotate = 15224; //激光旋转
e_mst_l2c_bc_change_max_power = 15225; //玩家等级改变
e_mst_l2c_bc_missile_fire = 15226; //导弹发射
e_mst_l2c_bc_debuginfo = 15227; //调试信息

//机器人
e_mst_c2l_robot_enter = 16000;
e_mst_l2c_robot_leave = 16100;

e_mst_clend_index = 20000;
}

```