### Command List

|  |  |  |  |
| --- | --- | --- | --- |
| 代碼 | 指令名稱 | 指令功能簡述 | 備註 |
|  |  | 創帳號(會員中心申請) | 需要包含幣別  美 中 歐 泰 越 印尼 韓 日 馬 |
|  |  | 登入(與會員中心同步Token) Client |  |
|  |  | 修改密碼(讓會員中心同步修改) Server |  |
|  |  | 抓帳 (注單資訊) to Club |  |
|  |  | 抓流水帳 (遊戲細節) to Club |  |
|  |  | 打碼量單日總量查詢(遊戲內) to Club |  |
|  |  | 轉額度 交易序號(會員中心提供)to Club 轉入轉出 |  |
|  |  | 查詢轉額度帳務(透過交易序號) 是否成功 |  |
|  |  | 審核保護 |  |
|  |  | 活動Api |  |
|  |  |  |  |
| G01 | enter\_game | 當有玩家申請進入遊戲 |  |
| G02 | shoot | 當有玩家一般射擊 |  |
| G03 | hit | 當有玩家一般射擊後擊中魚種 |  |
| G04 | feature\_shoot | 當有玩家小遊戲射擊 |  |
| G05 | feature\_hit | 當有玩家小遊戲射擊後擊中魚種 |  |
| G06 | resolve\_fish\_coin | 玩家結算 |  |
| G07 | sync | 玩家金幣校正 |  |
| G08 | jp\_winner | 查詢JP贏家ID(最近20筆) |  |
| G09 | win\_jp | Server通知JP中獎 | 只有拉到玩家收到 |
| G10 | jp\_pool | 更新JP獎金 |  |
| G11 | exit\_game | 有玩家離線 |  |
| G12 | shutdown | 遊戲伺服器(Game Server)廣播關機通知 | 全伺服器 |
| G13 | new\_fish | GS在遊戲新增魚種 | 單一桌次 |
| G14 | marquee | GS廣播遊戲資訊(跑馬燈) | 全伺服器 |
| G15 | scene\_alarm | GS廣播腳本預警 | 單一桌次 |
| G16 | fish\_alarm | GS廣播魚種預警 | 單一桌次 |
| G17 | pause | GS廣播遊戲暫停 | 單一桌次 |
| G18 | sense\_change | GS廣播通知遊戲結單強制節算 |  |

GAME\_IS\_PLAYING 多開 -5

VERSION\_NOT\_MATCH 版本錯誤-4

INVALID\_CMD\_FORMAT CMD格式錯誤 -3

EXCEPTION 例外狀況 -2

ERROR 錯誤 -1

OK=0

INVALID\_FISH 非法的魚(已死亡 沒有這個ID) 1

BULLET\_ID\_NOT\_EXIST 子彈不存在 2

BALANCE\_NOT\_ENOUGH 玩家餘額不足 3

INVALID\_FEATURE\_ID 小遊戲ID錯誤(未知ID /已經結束又再下Hit) 4

FEATURE\_FIRED 小遊戲已完成(已射擊 但又再射) 5

FEATURE\_NOT\_FIRED (小遊戲未發射 但下Hit) 6

BULLET\_ID\_EXIST 重複的子彈編號 7

### 指令說明G01(enter\_game)：

當玩家(player)要申請進入遊戲房間(game room)，須執行下列指令，進入的player可以得到伺服器(server)的回覆，同一場地的其他player可以得到此指令的廣播。

當在場的player的client發現有異狀，例如:該位子沒有player，但是卻射出子彈，player會自動發出此指令，並指定”refresh"更新當前桌次的房間資訊。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G01(Client發出)  enter\_game當玩家申請進入遊戲 | | | |
| **指令類別(sys):** | | game | |
| **指令名稱(cmd):** | | enter\_game | |
| **序號(sn):** | | -XXXXXXXX | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| udid | 裝置識別碼 |  |  |
| channel | 裝置平台 | 1,2,3 | IOS、Android、PC |
| publish\_ver | Client目前版本 | x.y.z |  |
| refresh | 重新取得資訊 | 0 | player是申請,設為0 |

Example：

{

"sys":"game",

"cmd":"enter\_game",

"sn":-4365817901339377661,

"data":

{

"udid": "xxikxie",

"channel": "1",

"publish\_ver":"1.0.1",

"refresh":"0",

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G01(Client發出refresh)  enter\_game當玩家申請進入遊戲 | | | |
| **指令類別(sys):** | | game | |
| **指令名稱(cmd):** | | enter\_game | |
| **序號(sn):** | | -XXXXXXXX | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| udid | 裝置識別碼 |  |  |
| channel | 裝置平台 | 1,2,3 | IOS、Android、PC |
| publish\_ver | Client目前版本 | x.y.z |  |
| refresh | 重新取得資訊 | 1 |  |

Example：

{

"sys":"game",

"cmd":"enter\_game",

"sn":-4365817901339377661,

"data":

{

"udid": "xxikxie",

"channel": "1",

"publish\_ver":"1.0.1",

"refresh":"1",

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G01(Server回覆申請玩家)  enter\_game當玩家申請進入遊戲 | | | |
| **回覆指令名稱(ret):** | | enter\_game | |
| **回覆指令標籤(re):** | |  | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| status | 錯誤代碼 | 0 ~ -X | 讓Client判斷data是否能用 |
| seat\_id | 座位代號 | 0~3 |  |
| shoot\_gap | Auto射擊間隔時間 | 0.33 | 單位 sec |
| dlg\_show\_chance | 魚出現隨機對話的機率 | 0~1 |  |
| dlg\_hit\_chance | 擊殺後魚語音的機率 | 0~1 |  |
| table\_id | 桌號 | 68827672 |  |
| user\_id | 玩家ID | 10000001 |  |
| nickname | 玩家暱稱 | MikeChang |  |
| background | 遊戲當前場景 | 0~4 |  |
| jp\_prize | 當前JP累積金額 | 12345678 |  |
| jp\_bet | 獲取JP的最低押注 | 5 |  |
| bet\_options | 下注選項 | [1,2,3,5,7,10] | 格式為陣列 |
| bullet\_options | 砲台選項 | 1~2 | "1"=集中砲,"2"=鎖定砲 |
| balance | 客戶遊戲中餘額 | 0~9999999 |  |
| table\_player\_info | 遊戲桌中所有玩家資訊 | **複合資料 備註** |  |
| **table\_player\_info 資料格式** | | | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| seat\_id | 座位代號 | 0~3 |  |
| user\_id | 玩家ID | 10000001 |  |
| nickname | 玩家暱稱 | MikeChang |  |
| balance | 客戶遊戲中餘額 | 0~9999999 |  |
| bullet\_type | 選擇砲台 | 1~2 | "1"=集中砲,"2"=鎖定砲 |
| last\_bet | 當前押注值 | 1~10 |  |

Example:

{

"ret":"enter\_game",

"data":

{

"re":

{

"status":0,

"shoot\_gap": 0.33,

"dlg\_show\_chance": 0.2,

"dlg\_hit\_chance": 0.2,

"table\_id": "68827672",

"user\_id": 10000001,

"nickname": "MikeChang",

"background": 0,

"jp\_prize": 12345678,

"jp\_bet": 5,

"bet\_options":[1,2,3,5,7,10],

"bullet\_options":["1","2"],

"balance":1111,

"table\_player\_info":

[

{

"seat\_id":0,

"user\_id":10000014,

"nickname": "berry",

"balance":1038,

"bullet\_type":"1",

"last\_bet": 10,

},

{

"seat\_id":1,

"user\_id":1000020,

"nickname": "duy",

"balance":2048,

"bullet\_type":"2",

"last\_bet": 5,

},

],

}

}

}

### 指令說明G02(shoot)：

當玩家(player)要射擊一發子彈(game room)，須執行下列指令，射擊的player可以得到伺服器(server)的回覆，同一場地的其他player可以得到此指令的廣播。

當射擊player的client發現有異狀，例如:該位子沒有player，但是卻射出子彈，player會自動發出此指令，並指定”cancel"更新當前桌次的房間資訊。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G01(Client發出)  Shoot一般子彈射擊 | | | |
| **指令類別(sys):** | | game | |
| **指令名稱(cmd):** | | shoot | |
| **序號(sn):** | | -XXXXXXXX | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| x | 點擊座標X |  | client上傳 |
| y | 點擊座標Y |  | client上傳 |
| bet | 押注金額 | 1~10 |  |
| bullet\_type | 子彈類型 | 1~2 |  |
| bullet\_id | 子彈序號 |  | client上傳 |

Example:

Request:

{

"sys":"game",

"cmd":"shoot",

“sn”:-14253896222,

"data":{

"x": 222, 點擊座標X

"y": 222, 點擊座標Y

"bet": 100, 押注金額

"bullet\_type": "0",

"bullet\_id": "1234",

},

}

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G01(Client發出取消)  Shoot一般子彈射擊 | | | |
| **指令類別(sys):** | | game | |
| **指令名稱(cmd):** | | shoot | |
| **序號(sn):** | | -XXXXXXXX | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| x | 點擊座標X |  |  |
| y | 點擊座標Y |  |  |
| bet | 押注金額 | 1~10 |  |
| bullet\_type | 子彈類型 | 1~2 |  |
| bullet\_id | 子彈序號 |  |  |

Example:

Request:

{

"sys":"game",

"cmd":"shoot",

“sn”:-14253896222,

"data":{

"x": 222, 點擊座標X

"y": 222, 點擊座標Y

"bet": 100, 押注金額

"bullet\_type": "0",

"bullet\_id": "1234",

},

}

### 指令說明G03(hit)：

請求時機：當玩家(player)子彈碰撞到魚觸發此指令，由伺服器(server)判斷此子彈所打中的魚是否給獎，若該子彈成功觸發給獎，server需回覆client是否給獎(有給魚id就是有)。

廣播：同一場地的其他player可以得到此指令的廣播，收到廣播的玩家(player)則表演魚死掉給獎的流程，子彈若沒打死任何魚則無需廣播。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G03(Client發出)  Hit(子彈擊中魚) | | | |
| **指令類別(sys):** | | game | |
| **指令名稱(cmd):** | | hit | |
| **序號(sn):** | | -XXXXXXXX | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| fish\_id | 魚的id, int陣列 | 數值陣列，最少一隻 |  |
| bullet\_type | 子彈類型 | 1~2 |  |
| bullet\_id | 子彈序號 | 最大12碼 |  |
| test\_kill | 數值0、1 | 0、1 | 非正式環境才有的功能 |
| test\_jackpot | 數值1 | 1 | 非正式環境才有的功能 |

Example:

Request:

{

"sys":"game",

"cmd":"hit",

"data":{

"bullet\_id":"123",

"bullet\_type":"1",

"fish\_id":[33,55,77],

"test\_kill"1, # 非正式環境, 必殺flag. 1=必殺, 0 = 必槓龜, 沒key=正常打擊

"test\_jackpot": 1, # 非正式環境, 1 = 必拉JP. (但bet必須大於minimal JP bet)

},

"sn":-4362053166504607737

}

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G03(Server回覆申請玩家)  Hit(子彈擊中魚) | | | |
| **回覆指令名稱(ret):** | | hit | |
| **回覆指令標籤(re)、廣播則為(fw):** | |  | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| Status | 錯誤代碼 | 0 ~ -X | 讓Client判斷data是否能用 |
| Bet | 押注金額 | 1000+ | 看該廳館砲的倍數 |
| Balance | 客戶遊戲中餘額 | 0~9999999 |  |
| bullet\_id | 子彈序號 | 最大12碼 | (由client產生) |
| kill\_info | 給獎的魚id陣列 | 0隻~n隻 | 回覆的陣列為空，代表都沒中獎 |
| kill\_info的陣列內容 |  |  |  |
| fish\_id | 魚的編id, int | 數值0~999999循環 |  |
| Win | 得分金額 | 數值，賠率x押注 |  |
| feature\_type | 特殊事件類別 | "1"=旋風, "2"=閃電, "3"=雷射, "4"=鑽頭, "5"=炸彈 | 特殊魚才有 |
| feature\_id | 特殊事件id | 字串，最大9碼 | 特殊魚才有 |

Response:

{

"ret":"hit",

"data":

{

"re": {

"status":0,

"bet": 100, 押注金額

"balance": 9640, 目前餘額

"bullet\_id":123,

"kill\_info": [

{

"fish\_id":123,

"win":500,

"feature\_type":"2", #string

"feature\_id":"1234567899",

},

{

"fish\_id":200,

"win":100,

},

],

}

}

### 指令說明G04(feature\_shoot)：

請求時機：當玩家(player)收到伺服器回傳成功擊殺的魚，當中有包含特殊魚種，該特殊魚種在client的表演後呼叫此指令，目的是拿回賠率。

廣播：同一場地的其他player可以得到此指令的廣播，收到廣播的玩家(player)則表演該特殊事件的發射方向(若武器有方向，ex.電磁砲和鑽頭砲)。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G04(Client發出)  feature\_shoot(特殊事件的發射) | | | |
| **指令類別(sys):** | | game | |
| **指令名稱(cmd):** | | feature\_shoot | |
| **序號(sn):** | | -XXXXXXXX | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| feature\_type | 特殊事件類別 | "1"=旋風, "2"=閃電, "3"=雷射, "4"=鑽頭, "5"=炸彈 |  |
| feature\_id | 特殊事件id | 字串，最大9碼 |  |
| X | 座標 | 螢幕範圍-寬(-n~n) |  |
| Y | 座標 | 螢幕範圍-高(-n~n) |  |

Example:

Request:

{

"sys":"game",

"cmd":"feature\_shoot",

"data":{ 發射資訊陣列

"feature\_type": "2", # "1"=旋風,"2"=閃電,"3"=雷射,"4"=鑽頭,"5"=炸彈

"feature\_id": "123456789",

"x": 222, 點擊座標X

"y": 222, 點擊座標Y

},

"cmd\_sn": ""

}

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G04(Server回覆申請玩家和廣播)  feature\_shoot(特殊事件的發射) | | | |
| **回覆指令名稱(ret):** | | feature\_shoot | |
| **回覆指令標籤(re)、廣播則為(fw):** | |  | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| Status | 錯誤代碼 | 0 ~ -X | 讓Client判斷data是否能用 |
| feature\_type | 特殊事件類別 | "1"=旋風, "2"=閃電, "3"=雷射, "4"=鑽頭, "5"=炸彈 |  |
| feature\_id | 特殊事件id | 字串，最大9碼 |  |
| Odds | 賠率 | 需參考賠率表range |  |
| X | 座標 | 螢幕範圍-寬(-n~n) |  |
| Y | 座標 | 螢幕範圍-高(-n~n) |  |

Example:

Response:

{

"ret":"feature\_shoot"

"data":{

"re":{

"status":0,

"feature\_type":"2" ,

"feature\_id": "1234567899",

"odds": 30, #

"x": 222, # 給其他玩家參考

"y": 222, # 給其他玩家參考

}

}

}

### 指令說明G05(feature\_hit)：

請求時機：當玩家(player)的特殊事件成功打到魚，就把擊中的魚送給伺服器(server)計算給獎。

廣播：同一場地的其他player可以得到此指令的廣播，收到廣播的玩家(player)則表演該特殊事件的給獎流程。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G05(Client發出)  feature\_hit(特殊事件的擊中) | | | |
| **指令類別(sys):** | | game | |
| **指令名稱(cmd):** | | feature\_hit | |
| **序號(sn):** | | -XXXXXXXX | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| fish\_id | 魚的id, int陣列 |  |  |
| feature\_type | 特殊事件類別 | "1"=旋風, "2"=閃電, "3"=雷射, "4"=鑽頭, "5"=炸彈 |  |
| feature\_id | 特殊事件id | 字串，最大9碼 |  |
| Stage | 字串 | ”0”、”1” | “0”：只要odds沒用完都直接死，鑽頭砲鑽到魚時會用。 “1”：正規化死亡率，讓傳上來的魚都有機會給獎，閃電、雷射、鑽頭的結尾、炸彈。 |

Example:

Request

{

"sys":"game",

"cmd":"feature\_hit",

"data":{

"fish\_id":[33,55,77], # twister不需要帶魚

"feature\_type":"X" # 1=TWISTER,2=LIGHTING,3=LASER,4=DRILL,5=BOMB

"feature\_id":"1234567899" #

"stage":"0", # "0": 丟上來只要odds沒用完都直接死; "1": 會正規化死亡率讓全部魚都有機會死 (最後結束時丟全部捕捉到的魚).

},

"sn":-4362053166504607737

}

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G05(Server回覆申請玩家和廣播)  feature\_hit(特殊事件的擊中) | | | |
| **回覆指令名稱(ret):** | | feature\_hit | |
| **回覆指令標籤(re)、廣播則為(fw):** | |  | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| status | 錯誤代碼 | 0 ~ -X | 讓Client判斷data是否能用 |
| balance | 玩家身上餘額double | 0~99999999999 |  |
| feature\_id | 特殊事件id | 字串，最大9碼 |  |
| remain\_odds | 沒用完的odds | 看賠率設定 | 應該可以不用回傳(client目前沒用到) |
| kill\_info | 給獎的魚的陣列 |  |  |
| kill\_info的內容： |  |  |  |
| fish\_id | 魚的編id, int | 數值0~999999循環 |  |
| win | 得分金額 | 數值，賠率x押注 |  |

Example:

Response

{

"ret":"feature\_hit",

"data":

{

"re": {

"status":0,

"balance": 9640, 目前餘額

"feature\_id":"1234567899"

"remain\_odds":35, #剩下的odds.

"kill\_info": [

{

"fish\_id":123,

"win":500,

},

{

"fish\_id":200,

"win":100,

},],

}

}

}

### 指令說明G06(resolve\_fish\_coin)：

請求時機：當玩家(player)結算時呼叫的指令。

廣播：無需廣播。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G06(Client發出)  resolve\_fish\_coin(結算) | | | |
| **指令類別(sys):** | | game | |
| **指令名稱(cmd):** | | resolve\_fish\_coin | |
| **序號(sn):** | | -XXXXXXXX | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| 無需參數 | | | |

Example:

Request

{

"sys":"game",

"cmd":"resolve\_fish\_coin",

"data":{},

"sn":-4362053166504607737

}

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G06(Server回覆申請玩家)  resolve\_fish\_coin(結算) | | | |
| **回覆指令名稱(ret):** | | resolve\_fish\_coin | |
| **回覆指令標籤(re):** | |  | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| status | 錯誤代碼 | 0 ~ -X | 讓Client判斷data是否能用 |
| user\_info | 玩家資訊 | 資料如下列 |  |
| user\_info內容 | | | |
| balance | 玩家餘額 |  |  |
| game\_session\_log | 注單 | 複合資料 | 請mike補充注單資料 |
| game\_session\_log內容 | | | |
| 請mike補充注單資料 |  |  |  |
|  |  |  |  |
|  |  |  |  |

Example:

Response:

"data":

{

"re": {

"status": 0, #0:成功，1:未知錯誤，-1:exception

"user\_info": {

"balance": 77810081,

},

//需跟mike確認注單內容

"game\_session\_log": [

{

"time":timestamp,

"bet":1000,

},

{

"time":timestamp,

"win":10000,

}

]

}

}

### 指令說明G07(sync)：

伺服器主動，同步玩家攜入金幣數量。

廣播：無需廣播。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G07(Server主動發送給該玩家)  sync(同步) | | | |
| **回覆指令名稱(cmd):** | | sync | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| diff\_coin | 差異的金額 | -n~n | 可為正或負(正常情況為正,除非有魚機+其他遊戲多開在玩) |

Example:

server cmd:

{

"cmd":"sync",

"data":{

"diff\_coin":1000 #可為正或負(正常情況為正,除非有魚機+其他遊戲多開在玩)

}

}

### 指令說明G08(jp\_winner)：

請求時機：當玩家(player)點開jackpot得主UI，client會發出這個請求，拿最近二十筆彩金得主資料。

廣播：無需廣播。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G08(Client發出)  jp\_winner(彩金得主清單) | | | |
| **指令類別(sys):** | | game | |
| **指令名稱(cmd):** | | jp\_winner | |
| **序號(sn):** | | -XXXXXXXX | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| 無需參數 | | | |

Example:

Request

{

"sys":"game",

"cmd":"jp\_winner",

"data":{},

"sn":-4362053166504607737

}

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G08(Server回覆申請玩家)  jp\_winner(彩金得主清單) | | | |
| **回覆指令名稱(ret):** | | jp\_winner | |
| **回覆指令標籤(re):** | |  | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| status | 錯誤代碼 | 0 ~ -X | 讓Client判斷data是否能用 |
| winner\_list\_limit | 一次給多少筆 | int 2位 |  |
| winner\_list | 彩金得主清單 | 資料如下列 |  |
| winner\_list內容 | | | |
| user\_id | 玩家id | 字串8碼 | 參考平台玩家id設計 |
| nick\_name | 玩家䁥稱 | 字串12碼 |  |
| jp\_coin | 彩金金額 | double | 由玩家累積 |

Example:

Response:

"data":

{

"re": {

"status": 0,

"winner\_list":[

{

"user\_id": 10000001,

"nick\_name": "nickname",

"jp\_coin": 123456789,

},

{

….

}

],

"winner\_list\_limit":20,

}

}

### 指令說明G09(win\_jp)：

伺服器主動，通知彩金得主。

廣播：無需廣播。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G09(Server主動發送給該玩家)  win\_jp(中彩金) | | | |
| **回覆指令名稱(cmd):** | | win\_jp | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| win | 彩金金額 | double | 拉中玩家共同累積的彩金金額 |

Example:

系統名稱: game

命令:win\_jp

server cmd:

{

"cmd":"win\_jp",

"data":{

"win":12345678,

}

### 指令說明G10(jp\_pool)：

伺服器廣播，通知同步更新彩金數值。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G10(Server主動通知)  jp\_pool(更新彩金池) | | | |
| **回覆指令名稱(cmd):** | | jp\_pool | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| prize | 目前累積的彩金金額 | double | 玩家共同累積的彩金金額 |
| jp\_bet | 拉彩金所需的押注 | 看那一館是幾分砲，至少要符合該館的押注 |  |

Example:

系統名稱: game

server cmd:

{

"cmd":"jp\_pool",

"data":{

"prize":12345678

"jp\_bet":10000

}

### 指令說明G11(exit\_game)：

伺服器主動，離桌的那個玩家不會收到。

廣播：通知該桌其他玩家(player)有玩家(player)離桌，。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G11(Server主動廣播給該桌其他玩家)  exit\_game(離桌) | | | |
| **回覆指令名稱(cmd):** | | exit\_game | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| seat\_id | 離桌的玩家座位id | 0~3 |  |
| user\_id | 離開的玩家id | 字串8碼 |  |

Example:

server broadcast:

{

"cmd":"exit\_game", #為server主動廣播該桌

"data":{

"seat\_id":1,

"user\_id":"10000001",

}

}

### 指令說明G12(shutdown)：

伺服器主動廣播，通知該伺服器準備關機。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G12(Server主動通知全伺服器)  shutdown(關機) | | | |
| **回覆指令名稱(cmd):** | | shutdown | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| min | 幾分鐘後關機 | 單位分鐘0~n |  |

Example:

命令:shutdown

{

"cmd":"shutdown",

"data":{

"min":1, # 幾分鐘後關機

}

### 指令說明G13(new\_fish)：

伺服器主動廣播，通知該桌產生新的魚。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G13(Server主動通知該桌)  new\_fish(產生魚) | | | |
| **回覆指令名稱(cmd):** | | new\_fish | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| status | 錯誤代碼 | 0 ~ -X | 讓Client判斷data是否能用 |
| background | 背景id | 參考出魚腳本中的背景Id |  |
| script\_name | 腳本檔案名稱(除錯用) | 字串(不超過22) |  |
| fish\_info | 魚資訊 | 複合資料 |  |
| fish\_info內容 | | | |
| fish\_id | 魚的id，int | 由server編號 |  |
| fish\_type | 魚的類型，int | 0, // 迦魶魚  1,// 小丑魚  2,// 鰈魚  3,// 河魨  4,// 獅子魚  5,// 比目魚  6,// 龍蝦  7,// 旗魚  8,// 章魚  9,// 燈籠魚  10,// 海龜  11,// 鋸齒鯊  12,// 蝠魟  13,// 巨大化--小丑魚  14,// 巨大化--鰈魚  15,// 巨大化--河魨  16,// 鯊魚  17,// 殺人鯨  18,// 座頭鯨  19,// 海王--火龍  20,// 海王--鱷魚  21,// 炸彈蟹  22,// 電磁蟹  23,// 電鑽蟹 |  |
| feature | int，特殊事件type | 0: 一般魚  1: 旋風魚  2: 閃電魚  3: 電磁砲 Type 22  4: 鑽頭砲Type 23  5: 炸彈Type 21 |  |
| **feature\_position** | **int，特殊魚所在位置** | **只有一次生成多隻魚才需要指定第幾隻為特殊魚，若1次1隻則為0** |  |
| x | 魚進場位置x軸 | -640~640 |  |
| y | 魚進場位置y軸 | -480~480 |  |
| o | 魚進場角度 | 0~360 |  |
| fish\_num | int，生成魚的數量 | 若1隻則為1 |  |
| fish\_array | int，魚陣id，需要定義魚陣型的腳本，事先放在client端 | 預設為0，不使用魚陣~魚陣腳本數量n |  |

Example:

table broadcast:

{

"cmd":"new\_fish", #為server主動廣播該桌

"data":{

"status":0,

"background":1,

"script\_name":"fishGameScript\_6.json"

"fish\_info":[{

"fish\_id":1,

"fish\_type":1,

"feature":1,

"feature\_position":0

"x":0,

"y":0,

"o":0,

"fish\_num":0,

"fish\_array":0, }]

}

}

### 指令說明G14(marquee)：

伺服器主動廣播，通知訊息公告，ex:中大獎通知。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G14(Server主動通知全伺服器)  marquee(跑馬燈公告訊息廣播) | | | |
| **回覆指令名稱(cmd):** | | marquee | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| msg | 訊息內容 | 不要是空字串 | client會依字數縮小字體 |

Example:

server cmd:

{

"cmd":"marquee",

"data":{

"msg":"test"

}

}

### 指令說明G15(boss\_alarm)：

伺服器主動廣播，對象是在同桌的玩家，通知即將有什麼特殊事件。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G15(Server主動通知該桌)  boss\_alarm(腳本預警) | | | |
| **回覆指令名稱(cmd):** | | boss\_alarm | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| id | 腳本id | 視腳本項目決定範圍 |  |

Example:

server cmd:

{

"cmd":"scene\_alarm",

"data":{

"id":0,

}

}

### 指令說明G16(fish\_alarm)：

伺服器主動廣播，對象是在同桌的玩家，通知即將有什麼魚種即將進場。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G16(Server主動通知該桌)  fish\_alarm(魚種預警) | | | |
| **回覆指令名稱(cmd):** | | fish\_alarm | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| id | 魚種id | 視魚種決定範圍 |  |

Example:

server cmd:

{

"cmd":"fish\_alarm",

"data":{

"id":0

}

}

### 指令說明G18(force\_resolve)：

伺服器主動廣播，對象是在同桌的玩家，遊戲結算。

指令內容表：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令G18(Server主動通知該桌)  force\_resolve(遊戲結算) | | | |
| **回覆指令名稱(cmd):** | | force\_resolve | |
| **資料名稱** | **資料說明** | **資料數值範圍** | **備註** |
| action | 數值0、1 | 0:觸發結算後，繼續遊戲  1:觸發結算後，結束遊戲回大廳 (1用於伺服器維護時使用 關閉後不再讓玩家進入) | 收到後玩家主動觸發G06 |

Example:

server cmd:

{

"cmd":"sense\_change",

"data":{

"action":1

}

}

ps1：

基礎規範

1. Client 觸發發出

“sys”:game (分開發出的指令別 遊戲相關game 登入相關login 帳務相關account)

“cmd”:enter\_game

“data”:

{

1. Server 依據Client觸發的回傳

“ret”:enter\_game

“data“:

{”re”

1. Server 依據Client觸發的廣播

“ret”

“data”:

{”fw”

1. Server主動的廣播

“cmd”:

“data”:參數