

WaWaJi Client-Server 信令交互流程

WaWaJiClient 与 WaWaJiServer 信令交互流程如下所示。

请注意：

1. 其中 1 ~ 6 节需要按照游戏流程组织。第 7 节用户信息更新，在游戏的整个生命周期中均可调用。
2. 以下每条信令必须包含字段：seq, cmd, data。建议 Client 端发送的 seq 采用递增的方式。
3. 预约之后的信令，必须包含 session_id 字段。
4. Client 回复 Server 时，需要把对应的 Server 的 seq 放在请求的 seq 字段中原样返回。
5. Client 收到 Server 的信令后，应马上发送与之对应的信令回复，否则 Server 端认为 Client 放弃游戏，会通知下一个预约用户开始上机。
6. 敏感信息（初始配置和游戏结果）加密方式使用 AES/CBC/PKCS5Padding 256位密钥，将向量内容放在密文前面一起做 Base64。

1 预约

信令名称	十进制值	传递方向	重试最大次数 (单位：次)	每次重试间隔 (单位：秒)
预约上机	513	Client— >Server	5	2
预约上机回复（内含预约上机结果）	272	Server— >Client	0	0

使用：

1. 用户点击预约，Client 发送 **预约上机**，直到收到 Server 返回的 **预约上机回复（内含预约上机结果）**。否则每隔 2 秒重试发送一次，最大重试次数为 5 次。

信令字段说明：

```

// 预约上机
{
  "seq": 1,
  "cmd": 513,
  "session_id": "xxx", // 如果 continue 为 1, 则必须带上上次游戏的 session_id, 否则
不填
  "data": {
    "time_stamp": 1515508734166,
    "continue": 0 // 0: 不继续玩, 1: 继续玩, 默认为 0
  }
}

// 预约上机回复 (内含预约上机结果)
{
  "seq": 1,
  "cmd": 272,
  "data": {
    "result": 1, // 0: 成功, 1: 失败
    "player": { // 接收信令的用户信息, 客户端用于检查是否是发送给自己的
      "id": "userId",
      "name": "userName"
    },
    "index": 1, // 当前排在预约队列中的多少位, 索引值从 1 开始
    "time_stamp": 1515508734166,
    "session_id": "xxx" // 此次抓娃娃的 session_id, 后续指定都要带上这个 session
_id
  }
}

```

2 取消预约

信令名称	十进制 值	传递方向	重试最大次数 (单 位: 次)	每次重试间隔 (单 位: 秒)
取消预约	514	Client— >Server	5	2
取消预约 回复	274	Server— >Client	0	0

使用:

1. 预约成功后，用户取消预约，Client 发送 **取消预约**，直到收到 Server 返回的 **取消预约回复**。否则每隔 2 秒重试发送一次，最大重试次数为 5 次。

信令字段说明：

```
// 取消预约
{
  "seq": 2,
  "cmd": 514,
  "session_id": "xxx",
  "data": {
    "time_stamp": 1515508734166
  }
}

// 取消预约回复
{
  "seq": 2,
  "cmd": 274,
  "session_id": "xxx",
  "data": {
    "time_stamp": 1515508734166
  }
}
```

3 上机

信令名称	十进制值	传递方向
准备上机	258	Server—>Client
回复收到准备上机	516	Client—>Server

使用：

1. 预约成功后，用户等待上机。轮到该用户上机时，Server 发送 **准备上机**，Client 需回复 **回复收到准备上机**。
2. 在没有收到“回复收到准备上机”或者“确认上机”上机之前，Server 会每隔一秒重发一次，最多发送 5 次。5 次之后，如果还是没有收到回复，则认为 Client 放弃上机，继续通知下一个预约用户上机。

信令字段说明:

```
// 准备上机
{
  "seq": 3,
  "cmd": 258,
  "session_id": "xxx",
  "data": {
    "player": { // 接收信令的用户信息, 客户端用于检查是否是发送给自己的
      "id": "userId",
      "name": "userName"
    },
    "game_time": 30,
    "time_stamp": 1515508734166
  }
}

// 回复收到准备上机
{
  "seq": 3,
  "cmd": 516,
  "session_id": "xxx",
  "data": {
    "time_stamp": 1515508734166
  }
}
```

4 确认上机

信令名称	十进制值	传递方向	重试最大次数 (单位: 次)	每次重试间隔 (单位: 秒)
上机或放弃上机	515	Client— >Server	5	2
回复收到上机选择	273	Server— >Client	0	0

使用:

1. 用户选择上机或放弃上机, Client 发送 **上机或放弃上机**, 直到收到 Server 返回的 **回复收到上机选择**。否则每隔 2 秒重试发送一次, 最大重试次数为 5 次。

2. 从收到通知准备上机至回复确认上机，此过程必须在 10s 内完成。

信令字段说明：

```
// 上机或放弃上机
{
  "seq": 4,
  "cmd": 515,
  "session_id": "xxx",
  "data": {
    "confirm": 1,      // 1: 确认上机, 0: 放弃上机
    "config": "encrypted-by-shared-secret-key", // 游戏初始化配置信息及付费校验信息。
    "time_stamp": 1515508734166
  }
}

// 回复收到上机选择
{
  "seq": 4,
  "cmd": 273,
  "session_id": "xxx",
  "data": {
    "result": 0,      // 0: 成功; 1: 格式无效; 2: 校验失败
    "time_stamp": 1515508734166
  }
}
```

请注意，上机或放弃上机信令中，必须携带加密后 config 字段。考虑到安全因素，该字段需由客户自己的业务后台生成，客户端从自己的业务后台获取。

客户业务后台对 config 加密前的明文格式如下：

```
{
  "game_config": {
    "game_time": 30,      // 游戏总时长
    "claw_power_grab": 67, // 表示抓起爪力(1-100)，指下爪时，抓住娃娃的爪力，建议这个值设置大一点
    "claw_power_up": 33,  // 表示到顶爪力(1-100)，指天车提起娃娃到 up_height 指定的高度后将使用该爪力值直至天车到达顶部
    "claw_power_move": 21, // 表示移动爪力(1-100)，指天车到达顶部后，移动过程中的爪力
    "up_height": 7        // 抓起高度 (0-10) 底部到顶部分成10份，爪子到达该值指定的高度时就会将爪力减小至到顶爪力
  },
  "authority_info": {
```

```

    "session_id": "xxx",    // 同信令中 session_id 值
    "confirm": 1,         // 同信令中 confirm 值
    "time_stamp": 1515508734166,    // 同信令中 time_stamp 值
    "custom_token": "xxx",    // 业务侧自定义鉴权信息，会在游戏结果加密串中带回，用于实现自定义校验，如支付信息等，该字段长度不要超过 300 个字符
    "room_id": "current_room_id" // 可选字段。当前房间 ID，用于验证是否是业务侧希望上机的房间
  }
}

```

1. 当 confirm 为 0 时（即放弃上机时），娃娃机不再校验 config 内容；
2. authority_info 中至少包含 session_id、confirm、time_stamp 三个字段，且与信令中的内容保持一致；
3. custom_token 字段内容长度`不能超过 300 个字符`，否则可能会导致信令发送失败；
4. custom_token 在游戏结束时，随结果加密串带回给 App 端；
5. room_id 当希望对上机进行强校验时，可以带上此字段，显示告诉娃娃机当前被授权用户应该上机的房间号，如校验不通过时，将不允许当前用户操作娃娃机。注意，如果不需要验证，则不要填该字段，而不是将此字段的内容置空（娃娃在检测到有此字段存在时，就会检查有效性）。
6. game_config 内容可以为空，但`必须设置为 {}`，而不能为空串或者没有该字段`。此时取默认参数初始化天车；
7. 当设置 game_config 值时，其中的抓力应满足： $claw_power_grab \geq claw_power_up \geq claw_power_move$ 。

客户业务后台对 config 加密算法的说明如下：

```

加密示例说明：
待加密内容：Content，即上述 config 加密前的明文内容
密钥(ZEGO 分配)：ServerSecret
向量：IV = Random.nextBytes(16)
加密后密文为：Base64.encode(IV + AES.encrypt(Content, ServerSecret, IV))

```

娃娃机加解密 go 语言源码请参考：[娃娃机加解密说明](#)

5 游戏操作

信令名称	十进制值	传递方向
左移	528	Client→Server
右移	529	Client→Server
前移	530	Client→Server
后移	531	Client→Server

下移（抓）	532	Client → Server
结束移动	533	Client → Server

使用：

1. 游戏操作指令，均不重试。
2. 下移指令，即抓娃娃。用户点击抓，或者游戏时间到了，Client 发送 **下移（抓）**，同时启动一个定时器，等待服务端返回游戏结果（即第 6 节表中的 **游戏结果** 指令）。
3. 发起移动指令，在结束移动前，需要每秒发送一次，并且 seq 与第一次的移动保持一致。

信令字段说明：

```
// 左移
{
  "seq": 5,
  "cmd": 528,
  "session_id": "xxx",
  "data": {
    "time_stamp": 1515508734166
  }
}

// 右移
{
  "seq": 6,
  "cmd": 529,
  "session_id": "xxx",
  "data": {
    "time_stamp": 1515508734166
  }
}

// 前移
{
  "seq": 7,
  "cmd": 530,
  "session_id": "xxx",
  "data": {
    "time_stamp": 1515508734166
  }
}

// 后移
{
```

```
"seq": 8,
"cmd": 531,
"session_id": "xxx",
"data": {
  "time_stamp": 1515508734166
}
}

// 下移 (抓)
{
  "seq": 9,
  "cmd": 532,
  "session_id": "xxx",
  "data": {
    "time_stamp": 1515508734166
  }
}

// 结束移动
{
  "seq": 10,
  "cmd": 533,
  "session_id": "xxx",
  "data": {
    "time_stamp": 1515508734166
  }
}
}
```

6 结果反馈

信令名称	十进制值	传递方向
游戏结果	260	Server—>Client
回复收到游戏结果	517	Client—>Server

使用:

1. 发送下抓指令后, 等待娃娃机 Server 返回游戏结果的时间应 **不少于 20 秒**, 超过此时间后, 则结束等待, 报错超时。
2. 如果 Client 在等待时间内, 收到了游戏回复, 则向 Server 发送 **回复收到游戏结果** reply。

信令字段说明:

```
// 游戏结果
{
  "seq": 11,
  "cmd": 260,
  "session_id": "xxx",
  "data": {
    "result": 1,           // 1: 中奖, 0: 未中奖
    "player": {          // 接收信令的用户信息, 客户端用于检查是否是发送给自己的
      "id": "userId",
      "name": "userName"
    },
    "time_stamp": 1515508734166,
    "encrypted_result": "" // 娃娃机将上机结果进行加密后的内容
  }
}

// 回复收到游戏结果
{
  "seq": 11,
  "cmd": 517,
  "session_id": "xxx",
  "data": {
    "continue": 0, // 0: 不继续玩, 1: 继续玩
    "time_stamp": 1515508734166
  }
}
```

上述 encrypted_result 加密前的 JSON 明文格式

```
{
  "session_id": "xxx",
  "result": 1,           // 1: 中奖, 0: 未中奖
  "player": {          // 接收信令的用户信息, 客户端用于检查是否是发送给自己的
    "id": "userId",
    "name": "userName"
  },
  "custom_token": "xxxxx", // 该 token 是在上机信令中的 config 中传给娃娃机的
  "time_stamp": 1515508734166,
  "room_id": "current_room_id" // 当前房间号, 业务侧可以根据此信息验证抓取结果的有效性
}
```

客户业务后台对 encrypted_result 解密算法的说明如下:

解密示例说明：

待解密密文：EncryptedContent

密钥（ZEGO 分配）：ServerSecret

解密后明文为：

```
PlainText = AES.decrypt(Content, ServerSecret, IV)
```

其中：

```
IV = Base64.decode(EncryptedContent).subString(0, 16)
```

```
Content = Base64.decode(EncryptedContent).subString(16)
```

娃娃机加解密源码请参考：[娃娃机加解密说明](#)

7 游戏信息更新

信令名称	十进制值	传递方向
获取游戏信息	518	Client→Server
回复获取游戏信息	275	Server→Client
游戏信息更新	257	Server→Client

使用：

- Client 向 Server 发送 [获取游戏信息](#)，需要在 Server 回复的 [回复获取游戏信息](#) 中获取到用户信息。
- Server 会在指定时刻（例如：用户退出房间、用户结束上机等），主动向 Client 发送 [游戏信息更新](#)，Client 可根据该指令内容，更新相关 UI。

信令字段说明：

```
// 获取游戏信息
{
  "seq": 12,
  "cmd": 518,
  "data": {
    "time_stamp": 1515508734166
  }
}

// 回复获取游戏信息
{
```

```

"seq": 12,
"cmd": 275,
"data": {
  "total": 0,
  "queue": [
    {
      "id": "userId",
      "name": "userName"
    },
  ],
  "player": { // 当前正在上机的玩家信息
    "id": "currentPlayerId",
    "name": "currentPlayerName",
    "left_time": 10
  },
  "game_time": 30,
  "time_stamp": 1515508734166
}
}

// 游戏信息更新
{
  "seq": 13,
  "cmd": 257,
  "data": {
    "total": 0,
    "queue": [
      {
        "id": "userId",
        "name": "userName"
      },
    ],
    "player": { // 当前正在上机的玩家信息
      "id": "currentPlayerId",
      "name": "currentPlayerName"
    },
    "time_stamp": 1515508734166
  }
}

```

8 关于连续玩的特别说明

目前发现 Client 端在实现连续玩时，没有遵照上述协议要求，从而出现一些问题。虽然 Server 端已经对此做了一些兼容。为充分利用设备资源，减少设备的不必空闲开销，后续 Server 端会移除这些兼容特性，完全按照上述协议来实现。特此就如何实现连续玩这一特性做统一说明：

1. 当游戏结束时，Server 端会发送游戏结果给 Client，Client 在收到“游戏结果”信令(260)后，马上发送“回复收到游戏结果”信令(517)，且 data 中的 continue 字段值为 1；否则 Server 端在发送 5 次后仍没有收到回复的话，会认为 Client 端已经下线，马上通知预约队列中的下一位玩家；
2. Client 端在发送完“回复收到游戏结果”信令(517)后，应该立即发送“预约”信令(513)，且 data 中的 continue 字段值为 1；
3. Client 端在收到“准备上机”信令(258)后，应马上发送“回复收到准备上机”信令(516)；
4. 上述第 3 步完成后，需要在 10S 内发送“上机或放弃上机”信令(515)，如果用户确认连续玩，data 中的 confirm 字段值为 1；否则为 0，取消上机。该过程，Server 端的最长等待时间为 15S；
5. 上述“回复收到游戏结果”&“预约”信令中，任意一条信令中的 continue 字段值为 0，则连续玩失败，进入普通预约流程；
6. 上述“预约”&“上机或放弃上机”信令均有可能失败，若 Server 发送给 Client 对应的 Reply 中包含错误信息，则连续玩失败；