

## Messaging protocol for HollywoodTV games

- Messaging protocol consists of JSON messages sent via HTTP POST requests from studio server to processing (remote) server.
- JSON messages have following fields for the remote server
  - **id**, string, unique string identifier for the draw within our system, currently we use sequential numbers for it.
  - **eventId**, number, unique sequential number identifier for the message
  - **gameId**, string, unique game name for the draw, LOTTO5 for this stream
  - **gameType**, string, game mode for the current game, specified for each stream separately
  - **type**, string, message type, messages with values other than “gameStatus” should be ignored (see below)
  - **status**, string, message should be ignored if it’s not one of the following values:
    - o Opened , bets are allowed
    - o Closed , bets are closed, draw itself has begun
    - o Drawing , single ball has been drawn/overridden.
    - o Completed, draw has been completed
    - o Validated, optional (see **hasValidation**), results have been validated and confirmed
    - o Cancelled , draw has been cancelled prior to beginning
    - o Void, draw has been cancelled after the beginning
  - **result**, for **ball games** it’s a string with selected balls, separated by hyphens, like “1-2-3-4-5-6”, can be empty or hold empty numbers like “1-2--3” for the cases when number is skipped, not selected. For **wheel games** it’s a string with wheel results, separated by hyphens. For **poker games** it contains JSON with following structure:

```
{
  "hands": [
    [ "H2", "C3" ],
    [ "C4", "C5" ],
    [ "H6", "S7" ],
    [ "S8", "S9" ],
    [ "D10", "CJ" ],
    [ "SQ", "DK" ]
  ],
  "flop": ["DA", "S2", "DJ" ],
  "turn": "HK",
  "river": "DQ"
}
```

Each card is represented with a code, first letter is suit (D, H, S, C), rest is value (2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A). Structure is empty at draw start and fills up step-by-step. Amount of hands is declared for each game separately.

- **drawDate**, string, UTC date and time for the draw beginning in “yyyy-MM-dd HH:mm:ss” format.
- **croupierName**, string, name of presenter on stage

- **presenterImageUrl**, string, transparent PNG photo of presenter on stage, intended for overlay purpose
- **videoUrl** - list of URLs for the live stream
  - o rtmp, string, live RTMP video stream
  - o hls, string, live HLS stream
- **recordUrl** - optional URL for HLS playlist, which has the record of the draw/round. Sent with **Completed** status.
- **hasValidation** - optional boolean flag, if present and true, this draw has separate completion and result validation - integrator system should hold winners calculation until Validation status has been received within a few seconds. **There is also a chance of Void status in very rare cases.**
- **initialState**, string, special optional field appearing in wheel games. Contains wheel results to which wheel(s) are reset in the beginning of the round. For multiple wheels results are separated by comma, like "31, X".
- **drawStep**, string, special optional field for poker games. Can be of **PreFlop, Flop, Turn, River** values or empty.
- **closeDate**, string, optional field with end date for betting step, contains UTC date and time in "yyyy-MM-dd HH:mm:ss" format.
  
- Messages are using queue, so they are sent one after another, queue waits for remote server response or timeout and won't remove message from queue until final status for the message is reached.
- Remote server is expected to use HTTP codes for message status determination. Following final codes are specified:
  - o 200, success, message is delivered and processed and will be removed from queue. **This code should be also used for ignored messages.**
  - o 422, failure, message is invalid, it's being removed from queue and show producer receives a notification or request to validate data. Validated data will be sent with separate **eventId**. This message is used when producer sends invalid data for round completion.
- Response body is expected to be empty. If not empty (for plain/text response) or contains "message" field (for application/json), this content will be sent to producer as notification.
- Messages are sent in the following order:
  - o Draw is opened,
  - o Draw is closed
  - o Balls/Results are being drawn
  - o Draw is complete/void
  - o End of the draw will send Opened message for the next draw.
    - For certain games we can setup sending Opened message for next draw prior to Closed message of current draw, this way bets can be accepted immediately after current draw is closed.
  - o Next draw is closed
  - o And so on, until final draw is complete. Final draw won't have a message for next draw open.
- Messages and responses are logged and stored for at least 3 months.