v.1.0.12

Third Party Messages API

Smart Move

Revision History

Date	Changes	Vesion	Author
May, 2016	First Release	v.1.0.0.	Roy Kornviets
June, 2016	1. Total format change 2. History examples added 3. Paragraph 2 Connection Details was added 4. Paragraph 6 Dictionaries and Naming conventions 5. API usage has been updated	v.1.0.1.	Marianna Reva
June,2016	1. operatorId, vipLevel and sessionCurrency fields were added to the p. 4.1 2. Initialize session example has been added to p.4 3. P 4.3 Tables list request has been added 4. P.5.9 new error example: “Maximum allowed sessions number exceeded” has been added	v.1.0.2.	Roy Kornvients, Marianna Reva
January, 2017	1. P4.1 Table active list response example has been updated with Roulette table 2. P.4.2 gameRoundId has been added to response message 3. Round Id has been added to Game History P4.6 4. 4.6.5 Game History Examples has been updated with Lottery and Keno Examples 5. P. 4.6.7 Error example has been updated 6. P.5.5, 5.6, 5.7, 5.8 have been updated. Round Id field has been added to the message. 7. P.5.8 Example of Keno has been added for game result messages 8. P.6.1 Game Ids codes has been updated with new game types 9. P6.2 table ids per studios has been updated 10. P. 5.1.15 Game specific Server to Client	v.1.0.3.	Roy Kornvients, Marianna Reva
April, 2017	1. P 4.1 Initialize session has been updated: IsActive parameter has been added to the table message, description of the method has been updated.	v.1.0.4.	Timur Rozovsky, Marianna Reva

June, 2017	1. P.4.6.7 Casino Hold'em Game History example has been added 2. P 5.1.5 RoundIndex param has been added to the roundStarted message. 3. P. 5.1.13 Dealer change, DealerId parameter has been added 4. P.5.3 Table Initialized, DealerId parameter has been added 5. P.5.15.1.2 End Game message has been updated. Card values of the round have been added 6. P.5.15.1.3 Baccarat Round Started message has been added 7. P.5.1.18 Game Result: Casino Hold'em game result has been added 8. P.6.1 Game Ids codes has been updated with new game Id values 9. P. 6.2 has been updated with the new table values 10. P.6.3 and 6.4 – card values and card suits have been added	v.1.0.5.	Timur Rozovsky, Marianna Reva
July, 2017	1. New mandatory field OperatorID has been added to the messages Seat Status Request (p4.2) and Table History (P.4.6).	v.1.0.6.	Roy Kornvients, Marianna Reva
July, 2017	1. Timestamp has been added to the messages initialize session (4.1) and round started (5.1.8) 2. P. 5.1.13.4 Cards Hands Message has been added 3. Additional comment for unlimited BJ has been added to p 5.1.11 and 5.1.12	v.1.0.7.	Roy Kornvients, Marianna Reva
August, 2017	1. P. 5.1.14.15 Cancel Last Card Message has been added	v.1.0.8.	Marianna Reva
November, 2017	1. P.2 API usage description has been updated. 2. P 4.2 Authenticate message has been added. 3. P 5.1.2 Authenticate token has been added 4. P.5.1.8 game results is updated with unlimited BJ examples 5. P 6.3 Table Ids has been updated; Unlimited BJ table Id has been added 6. P 6.4 Token PHP; Code Example has been added	v.1.0.9.	Roy Kornvients, Marianna Reva

July, 2018	1. vipLevel and sessionCurrency are returned with active table list response. See p. 4.2 Authenticate session, active table list, response message and response example 2. P. 3.1 Service URL has been updated	V1.0.10	Roy Kornvients, Marianna Reva
August, 2019	1. P. 4.7.8 Game History Examples has been updated with Three Card Poker/Teen Patty and Andar Bahar games examples	V1.0.11	Marianna Reva
September, 2019	1. P4.7.8 Game History Examples has been updated with Baccarat and Dragon Tiger new game result example 2. 9.6.2 Games per Studio has been removed 3. 4. P 6.1 Game Id codes has been updated	V1.0.12	Marianna Reva
March, 2022	1. Added the necessity of creating an endpoint for Authenticate Session call 2. Updated the WS URL 3. Removed Sede and KENO examples Updated JSON's	V1.1.0	Razvan Andrei

Contents

1. Document Overview ... 7
2. API Usage: ... 7
3. Connection Details ... 8
3.1 Service URL ... 8
4. Client to Server Messages ... 8
4.1 Initialize Session: ... 8
4.2 Authenticate Session ... 9
4.3 Seats Status Request (Only games that have table seats) ... 11
4.4 Table List Request ... 12
4.5 Register Session by Table Id: ... 12
4.6 Unregister Session by Table Id: ... 12
4.7 Table History ... 13
4.7.1 Baccarat Game History Example ... 13
4.7.2 Bet on Numbers Game History Example ... 14
4.7.3 Roulette Game History Example ... 14
4.7.4 Auto Roulette Game History Example ... 15
4.7.5 Casino Hold’em Game History Example ... 16
4.7.6 Three Card Poker/ Teen Patty Game History Example ... 18
4.7.7 Andar Bahar Game History Example ... 19
4.7.8 Game History Error Example: ... 20
5. Server to Client Messages: ... 21
5.1 Successful Registration ... 21
5.2 Authentication Token ... 21
5.3 Successful Un-registration: ... 21
5.4 Table Initialized: ... 22
5.5 Table Cancelled: ... 22
5.6 No More Bets: ... 22
5.7 Players Bets: ... 23
5.7.1 Blackjack: ... 23
5.7.2 Baccarat: ... 24
5.7.3 Roulette: ... 24
5.8 Game Result: ... 25
5.8.1 Blackjack: ... 25
5.8.2 Unlimited Blackjack: ... 27

5.8.3 Baccarat: 28
5.8.4 Dragon Tiger 29
5.8.5 Roulette 30
5.8.6 Casino Hold'em 31
5.8.7 Three Card Poker 33
5.8.8 Andar Bahar 34
5.9 Winning Ball 35
5.10 Round Started: 35
5.11 Round Cancelled: 35
5.12 Error Messages: 36
5.12.1 Wrong Table Id: 36
5.12.2 Table Inactive: 36
5.12.3 Maximum allowed sessions number exceeded 36
5.12.4 Seats Update (Only games that have table seats): 37
5.13 Seats Status (Only games that have table seats): 39
5.14 Dealer Changed: 40
5.15 Shuffle: 40
5.16 Game specific Server to Client Messages 41
5.16.1 Baccarat Card Message 41
5.16.2 End Round message 41
5.16.3 Baccarat Round Started: 42
5.16.4 Card Hands Message 42
5.16.5 Cancel Last Card Message 43
6. Dictionaries and Naming Conventions 44
6.1 Game Id codes 44
6.2 Card Values 45
6.3 Card Suits 45
6.4 Token PHP; Code Example 45

1. Document Overview

The Third-Party API was developed in order to allow third party operators to easily implement their own lobby for Ezugi games.

The Third-Party API sends messages to connected clients; constantly notifying them about the states of the Ezugi tables, available seats, limits per table and table history.

2. API Usage:

1. **In order to safely connect to the Web Socket service, the operator is required to create an endpoint. The usage of this endpoint is explained in point number 3.**
2. Connect to the **gameNotifications** web socket on the desired Ezugi game server (Example: wss://engineint.tableslive.com/GameServer/gameNotifications). After connecting to the game server, the connecting client will have 3 minutes to initialize the connection. If the connection wasn't initialized, the client will be disconnected.
3. Initialize the connection through the **InitializeSession** message. The connection will be initialized only if the connecting client didn't exceed the allowed connection limit for the client's operator. **After the initialization the Ezugi server will send a temporary token to the connecting server's callback URL. When receiving the token the connecting server should send an AuthenticateSession message that will authenticate the current session.** If the authentication process was successful, the connecting server will receive an active tables list. The connecting server will be also notified on tables that will be closed and open and on dealer changes on the table.
4. Register to a wanted table through a **RegisterSessionByTableId** message to receive the relevant messages from a particular table.

3. Connection Details

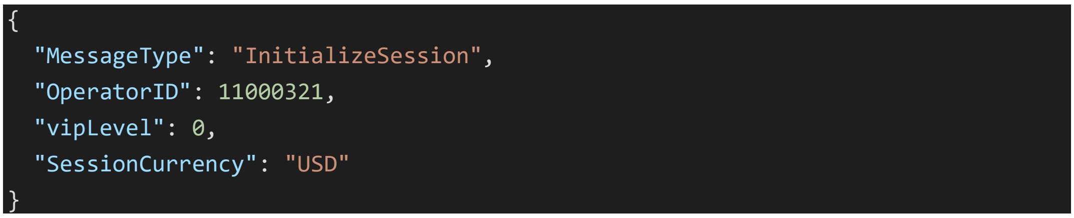
3.1 Service URL

Integration - wss://engineint.tableslive.com/GameServer/gameNotifications

4. Client to Server Messages

4.1 Initialize Session:

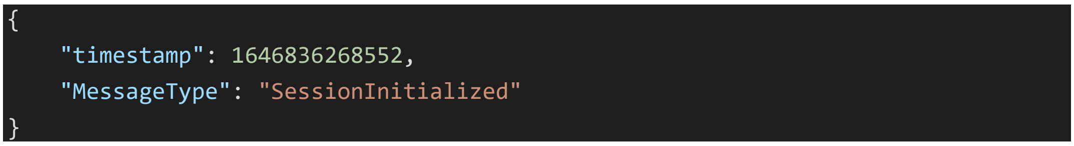
Request Message:



```
{
  "MessageType": "InitializeSession",
  "OperatorID": 11000321,
  "vipLevel": 0,
  "SessionCurrency": "USD"
}
```

The message will initialize the opened websocket session (unless there are too many websocket connections already open). If the initialization was successful, the client will receive an active tables list. The client will be also notified on tables that will be closed and open and on dealer changes on the table.

Response Message:

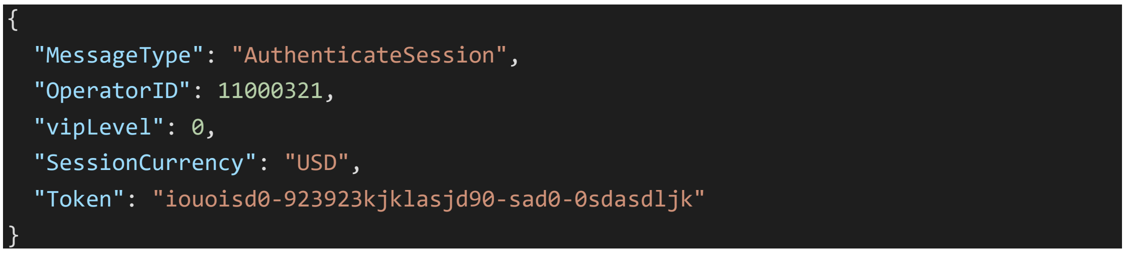


```
{
    "timestamp": 1646836268552,
    "MessageType": "SessionInitialized"
}
```

4.2 Authenticate Session

The message will authenticate the current server session and allow the connecting server to register to tables and receive game events.

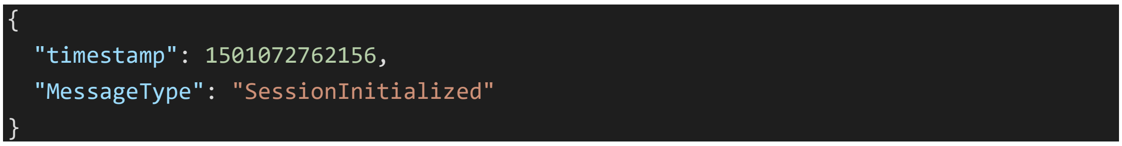
Message Example:



```
{
  "MessageType": "AuthenticateSession",
  "OperatorID": 11000321,
  "vipLevel": 0,
  "SessionCurrency": "USD",
  "Token": "iouoisd0-923923kjklasjd90-sad0-0sdasdljk"
}
```

Response:

Session Initialized:



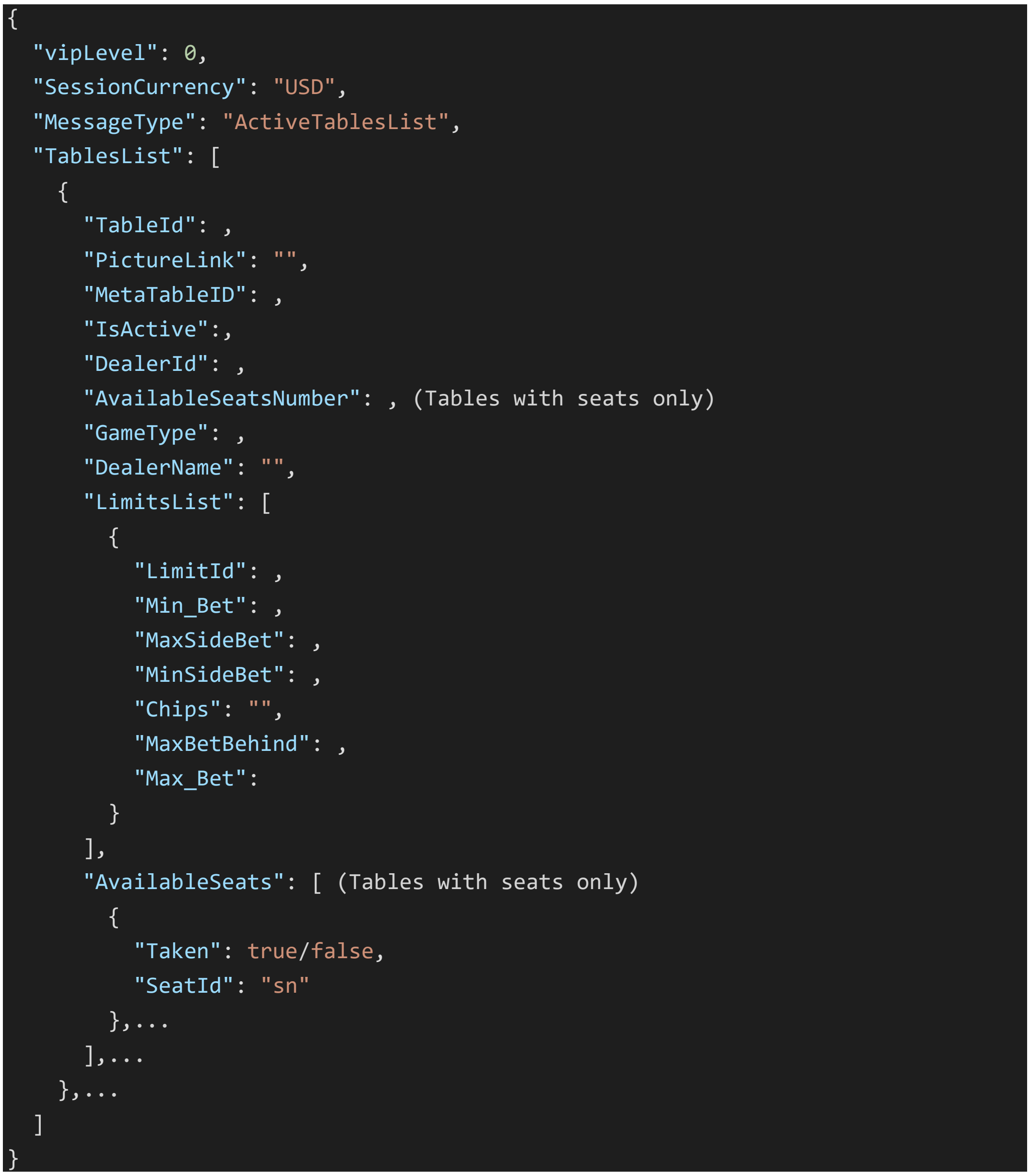
```
{
  "timestamp": 1501072762156,
  "MessageType": "SessionInitialized"
}
```

The message will notify the initializing client that the initialization completed successfully.

Active Table List:

The message delivers the client a list of current tables. All active tables will be sent with IsActive:true, dealer name and the picture link. The message will be sent to the client after the connection has been established.

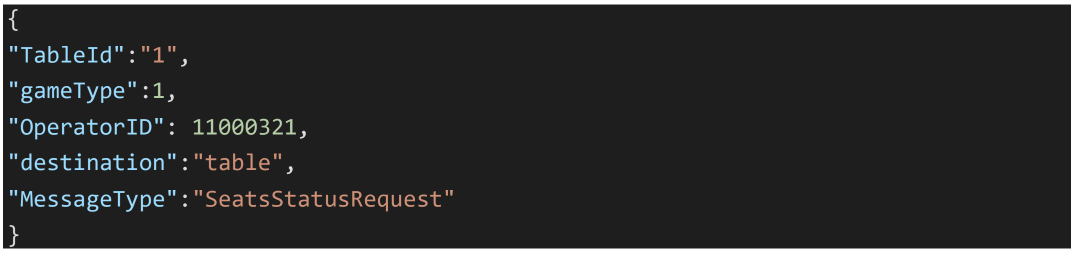
Response Message:



```
{
  "vipLevel": 0,
  "SessionCurrency": "USD",
  "MessageType": "ActiveTablesList",
  "TablesList": [
    {
      "TableId": ,
      "PictureLink": "",
      "MetaTableID": ,
      "IsActive":,
      "DealerId": ,
      "AvailableSeatsNumber": , (Tables with seats only)
      "GameType": ,
      "DealerName": "",
      "LimitsList": [
        {
          "LimitId": ,
          "Min_Bet": ,
          "MaxSideBet": ,
          "MinSideBet": ,
          "Chips": "",
          "MaxBetBehind": ,
          "Max_Bet":
        }
      ],
      "AvailableSeats": [ (Tables with seats only)
        {
          "Taken": true/false,
          "SeatId": "sn"
        },...
      ],...
    },...
  ]
}
```

4.3 Seats Status Request (Only games that have table seats)

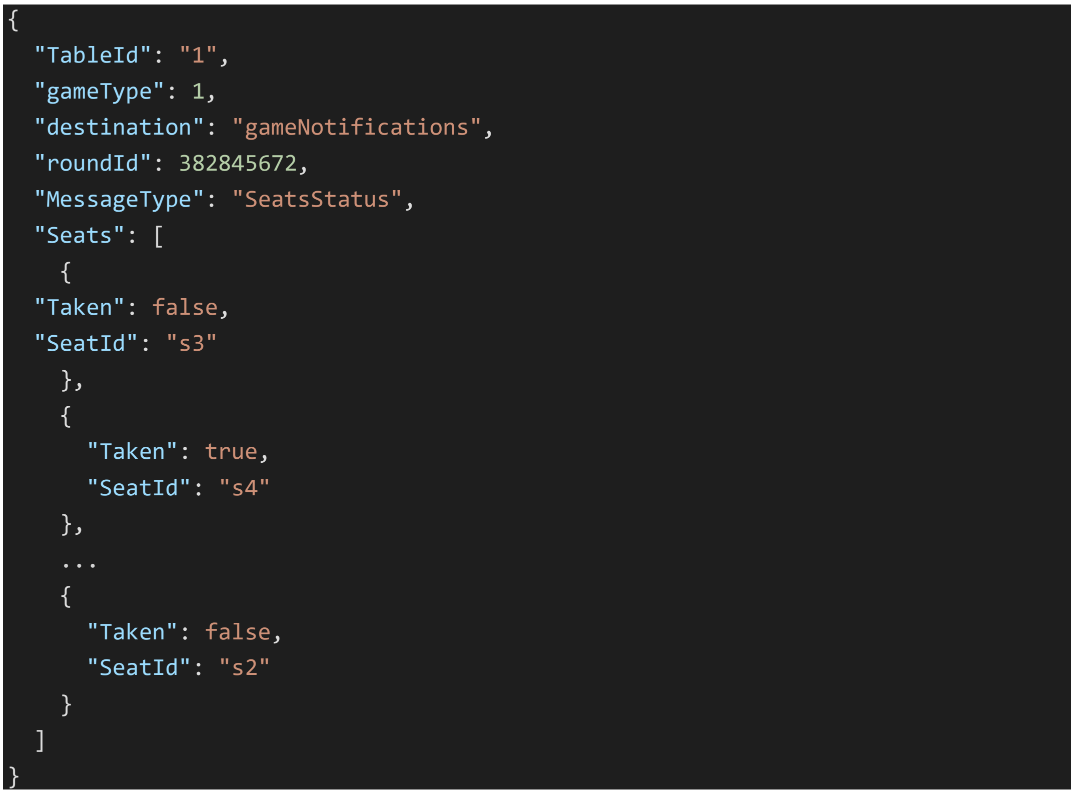
Request Message:



```
{
"TableId":"1",
"gameType":1,
"OperatorID": 11000321,
"destination":"table",
"MessageType":"SeatsStatusRequest"
}
```

The message will be sent from the client on order to request seats status of the results from a given table. The server will reply with a SeatsStatus message.

Response Example:



```
{
  "TableId": "1",
  "gameType": 1,
  "destination": "gameNotifications",
  "roundId": 382845672,
  "MessageType": "SeatsStatus",
  "Seats": [
    {
  "Taken": false,
  "SeatId": "s3"
    },
    {
      "Taken": true,
      "SeatId": "s4"
    },
    ...
    {
      "Taken": false,
      "SeatId": "s2"
    }
  ]
}
```

4.4 Table List Request

Request Message:

```
{
  "MessageType": "TablesListRequest",
  "OperatorID": 11000321,
  "vipLevel": 0,
  "SessionCurrency": "USD"
}
```

The message will be sent from the client to the server in order to request the current active tables list. The server will reply with an ActiveTablesList message.

4.5 Register Session by Table Id:

Request Message:

```
{
"TableId":"100",
"MessageType":"RegisterSessionByTableId",
 "OperatorID":11000321
}
```

The message will be sent from client to the server and will register the opened session on a given table. After registering the session, all the relevant messages from the table will be pushed via the session to the client.

Response Example:

```
{
  "TableId": "100",
  "MessageType": "SuccessfulRegistration"
}
```

4.6 Unregister Session by Table Id:

Request Message:

```
{
"TableId":"100",
"MessageType":"UnregisterSessionByTableId",
"OperatorID": 11000321
}
```

The message will be sent from the client in order to unregister a session from a given table. After unregistering the client will stop receiving messages from that table.

Request Example:

```
{
"TableId":"100",
"MessageType":"UnregisterSessionByTableId",
"OperatorID": 11000321
}
```

Response Example:

```
{
  "TableId": "100",
  "MessageType": "SuccessfulUnegistration",
  "OperatorID": 1100321
}
```

4.7 Table History

Request Message:

```
{
  "TableId": "100",
  "gameType": 2,
  "OperatorID":11000321,
  "destination": "table",
  "MessageType": "HistoryRequest"
}
```

The message will be sent from the client on order to request a history of the results from a given table. The server will reply with a TableHistory message.

The message delivers the client the results history of the table.

4.7.1 Baccarat Game History Example

History Request:

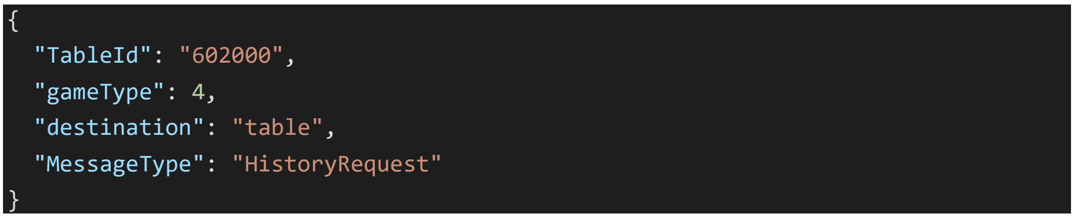
```
{
  "TableId": "100",
  "gameType": 2,
  "OperatorID": 11000321,
  "destination": "table",
  "MessageType": "HistoryRequest"
}
```

Table History Response:

```
{
  "TableId": "100",
  "gameType": 2,
  "destination": "gameNotifications",
  "History": [
    {
      "isPlayerPair": false,
      "isBankerPair": false,
      "WinningHand": "Banker",
      "CardHandValue": 8
    }
  ],
  "MessageType": "TableHistory"
}
```

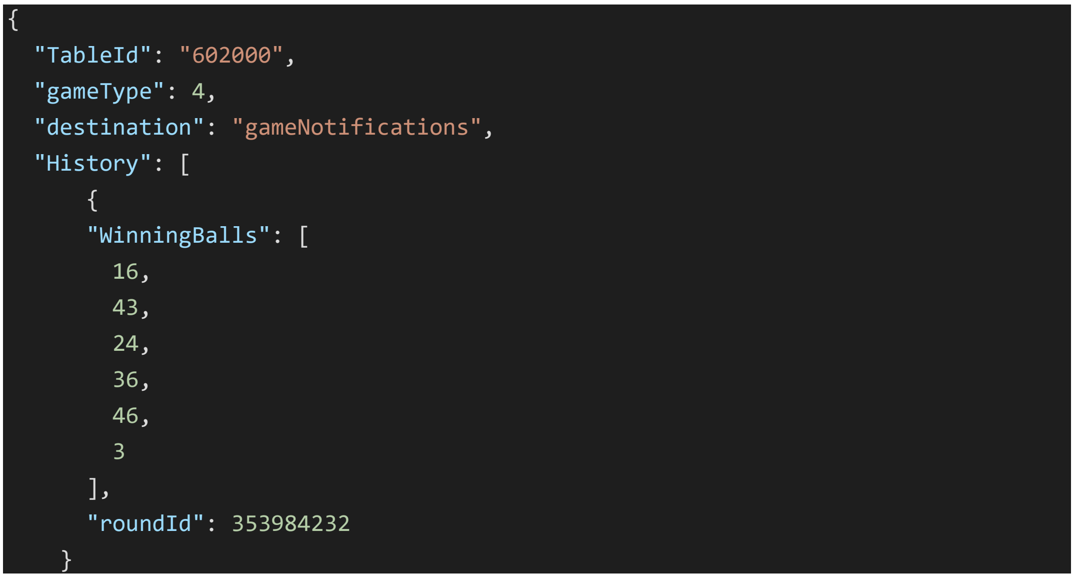
4.7.2 Bet on Numbers Game History Example

History Request:



```
{
  "TableId": "602000",
  "gameType": 4,
  "destination": "table",
  "MessageType": "HistoryRequest"
}
```

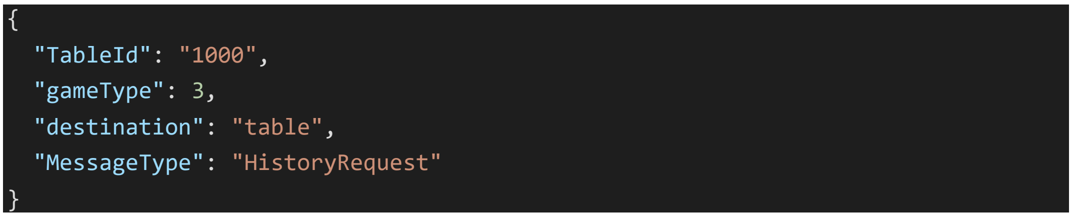
History Response:



```
{
  "TableId": "602000",
  "gameType": 4,
  "destination": "gameNotifications",
  "History": [
      {
      "WinningBalls": [
        16,
        43,
        24,
        36,
        46,
        3
      ],
      "roundId": 353984232
    }
```

4.7.3 Roulette Game History Example

History Request



```
{
  "TableId": "1000",
  "gameType": 3,
  "destination": "table",
  "MessageType": "HistoryRequest"
}
```

History Response

```
{
  "TableId": "1000",
  "gameType": 3,
  "destination": "gameNotifications",
  "History": [
    {
      "WinningNumber": 7,
      "roundId": 3410364308
    },
    {
      "WinningNumber": 9,
      "roundId": 3410364318
    }
  ],
  "MessageType": "TableHistory"
}
```

4.7.4 Auto Roulette Game History Example

History Request:

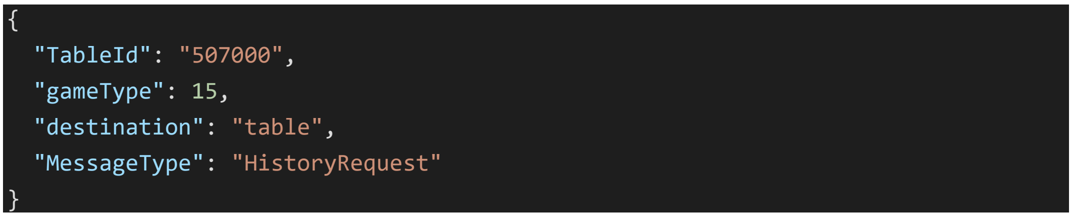
```
{
  "TableId": "5001",
  "gameType": 7,
  "destination": "table",
  "MessageType": "HistoryRequest"
}
```

History Response:

```
{
  "TableId": "5001",
  "gameType": 7,
  "destination": "gameNotifications",
  "History": [
    3,
    11,
],
  "MessageType": "TableHistory"
}
```

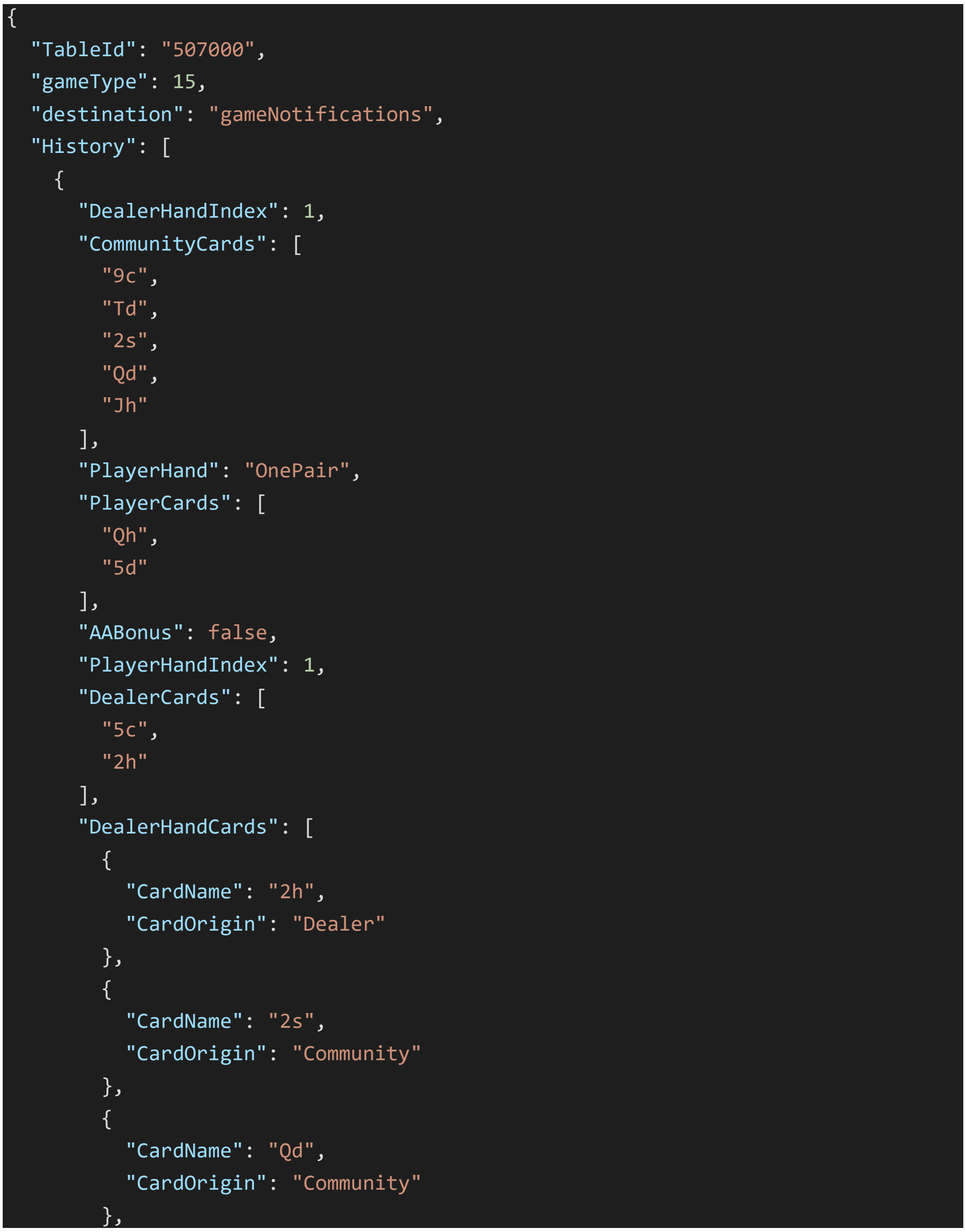
4.7.5 Casino Hold'em Game History Example

History Request:

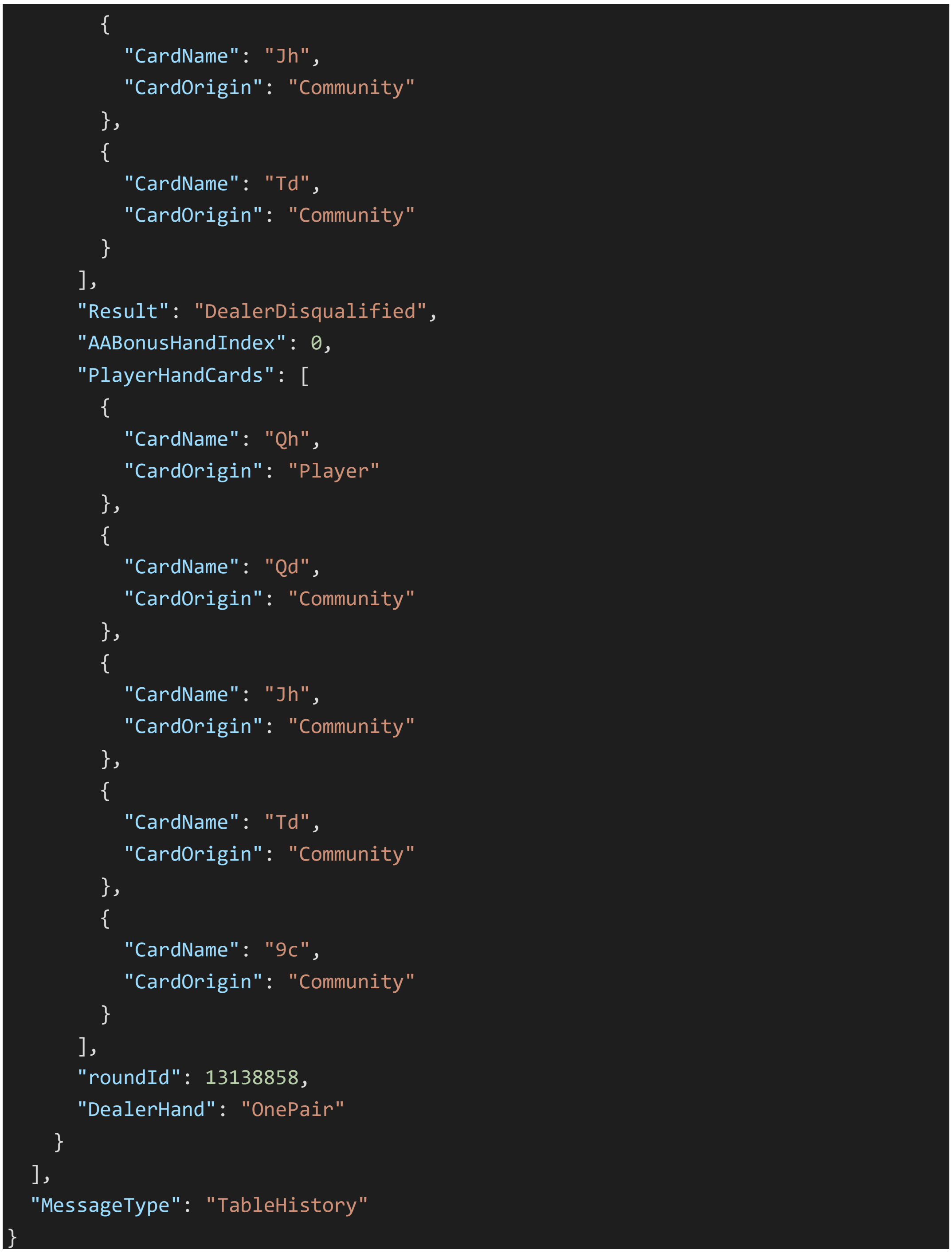


```
{
  "TableId": "507000",
  "gameType": 15,
  "destination": "table",
  "MessageType": "HistoryRequest"
}
```

History Response:



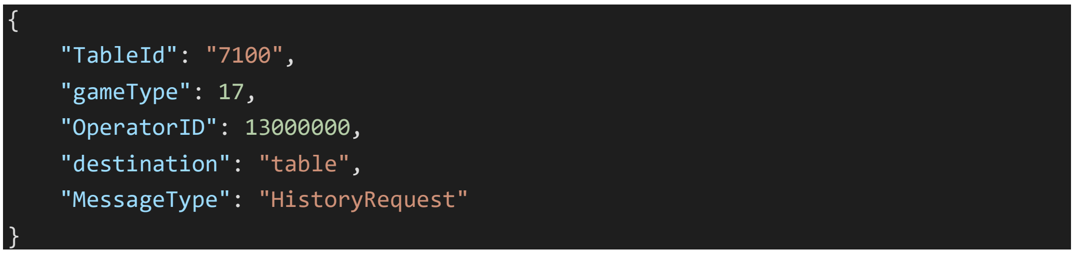
```
{
  "TableId": "507000",
  "gameType": 15,
  "destination": "gameNotifications",
  "History": [
    {
      "DealerHandIndex": 1,
      "CommunityCards": [
        "9c",
        "Td",
        "2s",
        "Qd",
        "Jh"
      ],
      "PlayerHand": "OnePair",
      "PlayerCards": [
        "Qh",
        "5d"
      ],
      "AABonus": false,
      "PlayerHandIndex": 1,
      "DealerCards": [
        "5c",
        "2h"
      ],
      "DealerHandCards": [
        {
          "CardName": "2h",
          "CardOrigin": "Dealer"
        },
        {
          "CardName": "2s",
          "CardOrigin": "Community"
        },
        {
          "CardName": "Qd",
          "CardOrigin": "Community"
        },
```



```
        {
          "CardName": "Jh",
          "CardOrigin": "Community"
        },
        {
          "CardName": "Td",
          "CardOrigin": "Community"
        }
      ],
      "Result": "DealerDisqualified",
      "AABonusHandIndex": 0,
      "PlayerHandCards": [
        {
          "CardName": "Qh",
          "CardOrigin": "Player"
        },
        {
          "CardName": "Qd",
          "CardOrigin": "Community"
        },
        {
          "CardName": "Jh",
          "CardOrigin": "Community"
        },
        {
          "CardName": "Td",
          "CardOrigin": "Community"
        },
        {
          "CardName": "9c",
          "CardOrigin": "Community"
        }
      ],
      "roundId": 13138858,
      "DealerHand": "OnePair"
    }
  ],
  "MessageType": "TableHistory"
}
```

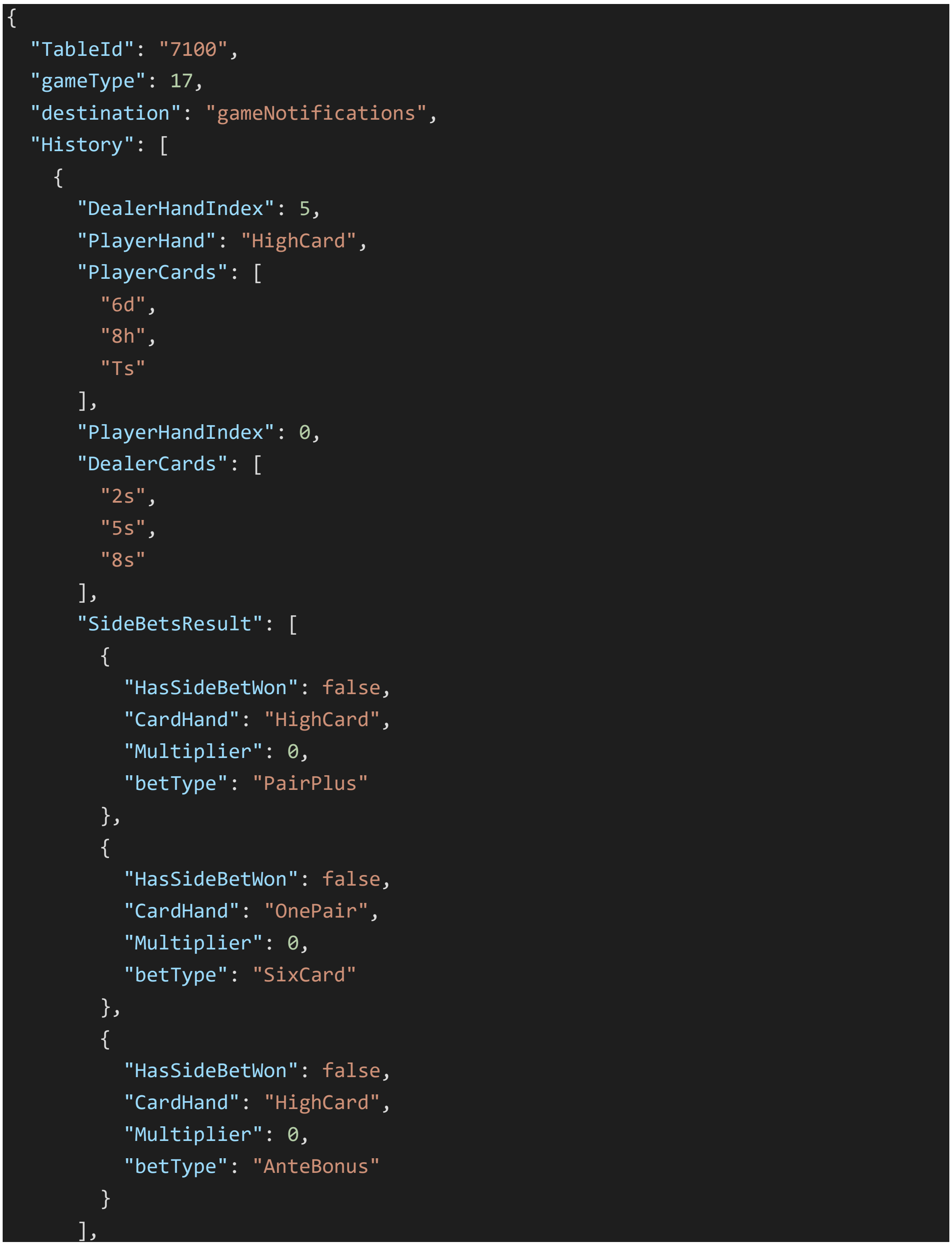
4.7.6 Three Card Poker/ Teen Patty Game History Example

History Request:

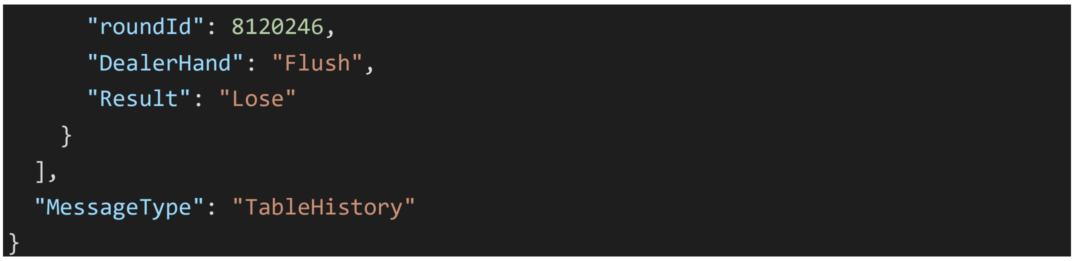


```
{
    "TableId": "7100",
    "gameType": 17,
    "OperatorID": 13000000,
    "destination": "table",
    "MessageType": "HistoryRequest"
}
```

History Response:



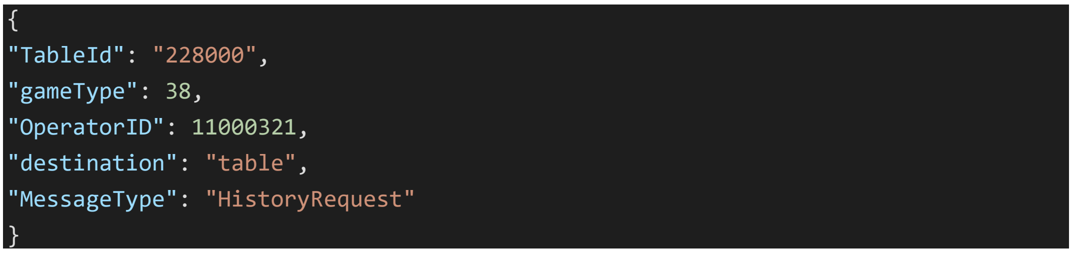
```
{
  "TableId": "7100",
  "gameType": 17,
  "destination": "gameNotifications",
  "History": [
    {
      "DealerHandIndex": 5,
      "PlayerHand": "HighCard",
      "PlayerCards": [
        "6d",
        "8h",
        "Ts"
      ],
      "PlayerHandIndex": 0,
      "DealerCards": [
        "2s",
        "5s",
        "8s"
      ],
      "SideBetsResult": [
        {
          "HasSideBetWon": false,
          "CardHand": "HighCard",
          "Multiplier": 0,
          "betType": "PairPlus"
        },
        {
          "HasSideBetWon": false,
          "CardHand": "OnePair",
          "Multiplier": 0,
          "betType": "SixCard"
        },
        {
          "HasSideBetWon": false,
          "CardHand": "HighCard",
          "Multiplier": 0,
          "betType": "AnteBonus"
        }
      ],
```



```
      "roundId": 8120246,
      "DealerHand": "Flush",
      "Result": "Lose"
    }
  ],
  "MessageType": "TableHistory"
}
```

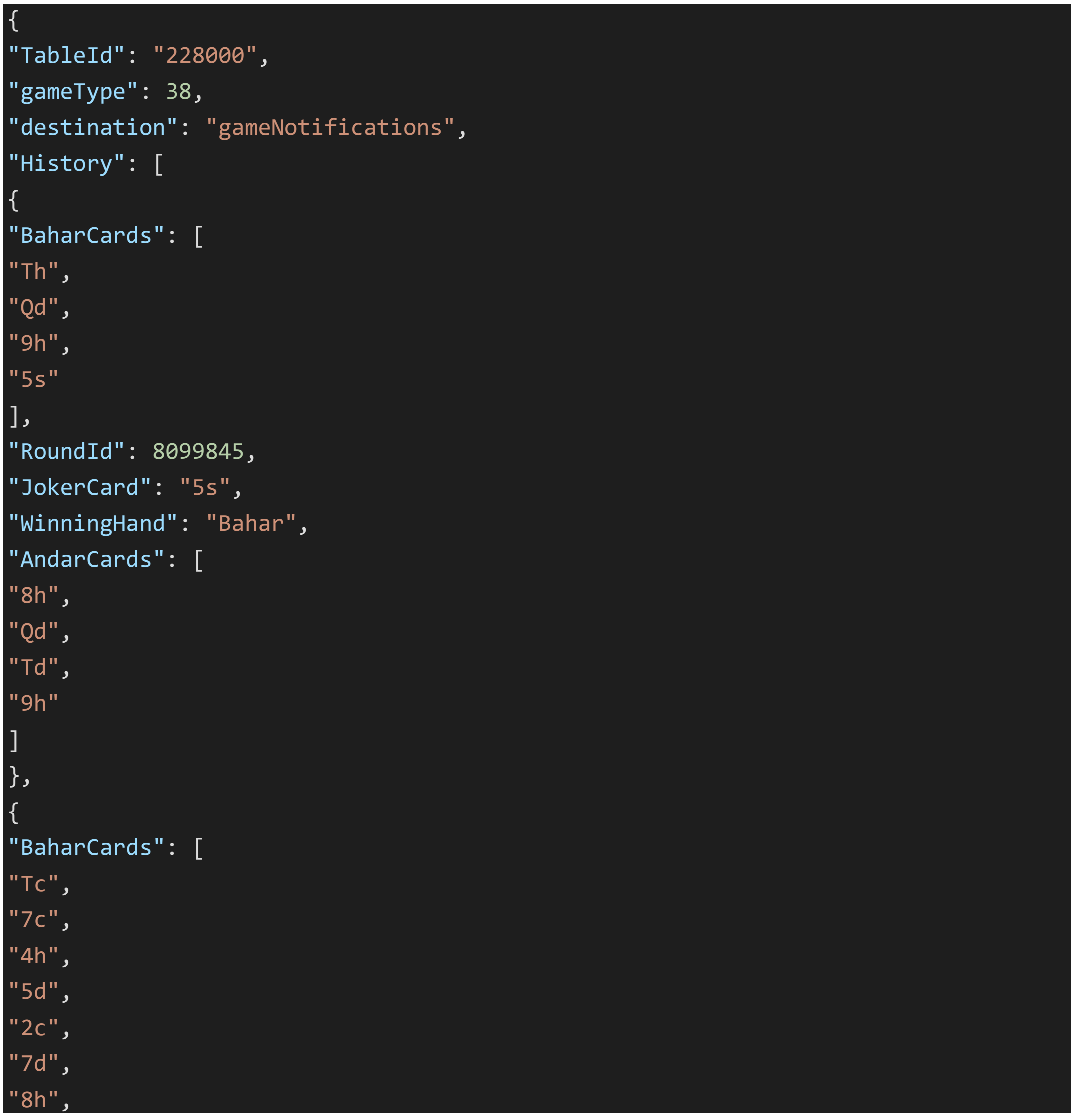
4.7.7 Andar Bahar Game History Example

History Request:



```
{
"TableId": "228000",
"gameType": 38,
"OperatorID": 11000321,
"destination": "table",
"MessageType": "HistoryRequest"
}
```

History Response:



```
{
"TableId": "228000",
"gameType": 38,
"destination": "gameNotifications",
"History": [
{
"BaharCards": [
"Th",
"Qd",
"9h",
"5s"
],
"RoundId": 8099845,
"JokerCard": "5s",
"WinningHand": "Bahar",
"AndarCards": [
"8h",
"Qd",
"Td",
"9h"
]
},
{
"BaharCards": [
"Tc",
"7c",
"4h",
"5d",
"2c",
"7d",
"8h",
"9c",
"4s",
"Kd"
],
"RoundId": 8099851,
"JokerCard": "Kd",
"WinningHand": "Bahar",
"AndarCards": [
"9s",
"4h",
"3d",
"8s",
"4s",
"5s",
"9c",
"6h",
"Qc",
"4d"
]
}
],
"MessageType": "TableHistory"
}
```

4.7.8 Game History Error Example:

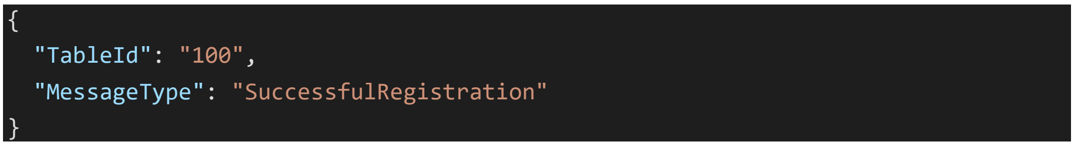
```
{
  "ErrorDescription": "Inactive table",
  "ErrorCode": 30055,
  "MessageType": "Error"
}
```

5. Server to Client Messages:

5.1 Successful Registration

The message will be sent if a registration to a specified table was successful as a response for the register session by table id message.

Response Message Example:

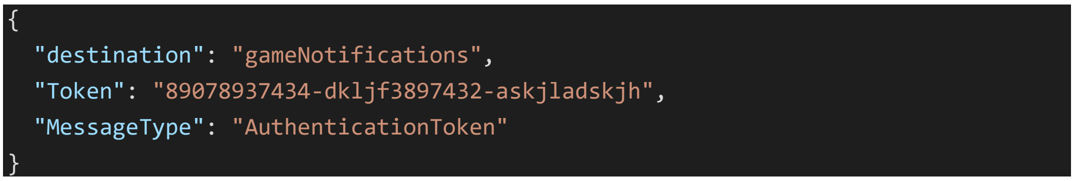


```
{
  "TableId": "100",
  "MessageType": "SuccessfulRegistration"
}
```

5.2 Authentication Token

The message will deliver the client an authentication token that should be sent back to Ezugi in order to authenticate the current session.

Response Message Example:

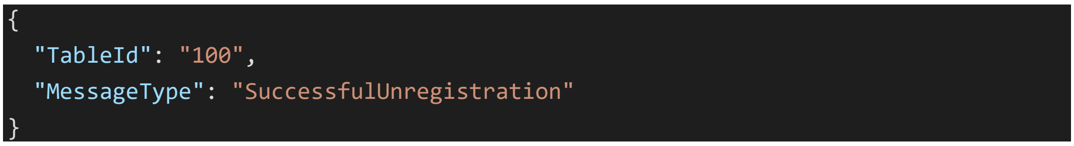


```
{
  "destination": "gameNotifications",
  "Token": "89078937434-dkljf3897432-askjladskjh",
  "MessageType": "AuthenticationToken"
}
```

5.3 Successful Un-registration:

The message will be sent if an un-registration to a specified table was successful as a response for the unregister session by table id message.

Response Message Example:



```
{
  "TableId": "100",
  "MessageType": "SuccessfulUnregistration"
}
```

5.4 Table Initialized:

The message notifies the clients that a table was opened.

The message will deliver the clients the table id, dealer name and the game type of the game on the table.

Response Message Example:

```
{
  "gameType": 1,
  "PictureLink":
"/LiveCasinoNew/BO/BizConfigs/Casino/Resources/Dealers/Images/Giannina_20.png",

  "destination": "gameNotifications",
  "tableId": "1",
   "DealerId": "20",
  "DealerName": "Giannina",
  "MessageType": "TableInitialized"
}
```

5.5 Table Cancelled:

The message notifies the clients that a table was cancelled.

Response Message Example:

```
{
  "gameType": 3,
  "destination": "gameNotifications",
  "tableId": "1000",
  "MessageType": "TableCancelled"
}
```

5.6 No More Bets:

The message notifies the client that bets are closed on the table.

Response Message Example:

```
{
  "TableId": "6000",
  "gameType": 6,
  "destination": "gameNotifications",
  "roundId": 10366192,
  "MessageType": "NoMoreBets"
}
```

5.7 Players Bets:

Message Example:

```
{
  "TableId": "",
  "gameType":,
  "PlayersBets":,
  "destination": "gameNotifications",
  "roundId":,
  "MessageType": "PlayersBets"
}
```

The message will deliver a list of players bets to the client after server has done processing the players bets.

Examples:

5.7.1 Blackjack:

```
{
  "TableId": "1",
  "gameType": 1,
  "PlayersBets": [
    {
      "BetBehindBetCount": 0,
      "BetBehindBetAmount": 0,
      "OperatorID": "10116001",
      "BetsList": [
        {
          "BetName": "RegularBet",
          "BetAmount": 10
        }
      ],
      "SeatId": "s4",
      "Nickname": "sashotest3"
    }
  ],
  "destination": "gameNotifications",
  "roundId": 10417370,
  "MessageType": "PlayersBets"
}
```

5.7.2 Baccarat:

```
{
  "TableId": "100",
  "gameType": 2,
  "PlayersBets": [
    {
      "BetsList": {
        "SideBets": {
          "PlayerPair": 10
        },
        "PlayerBet": 10
      },
      "Nickname": "CharlesDarwin"
    }
  ],
  "destination": "gameNotifications",
   "roundId": 10417377,
  "MessageType": "PlayersBets"
}
```

5.7.3 Roulette:

```
 {
  "TableId": "1000",
  "gameType": 3,
  "PlayersBets": [
    {
      "BetsList": [
        {
          "betValue": 10,
          "index": "100",
          "group": "1"
        }
      ],
      "Nickname": "AlbertEinstein"
    }
  ],
  "destination": "gameNotifications",
  "roundId": 10417390,
  "MessageType": "PlayersBets"
}
```

5.8 Game Result:

Message:

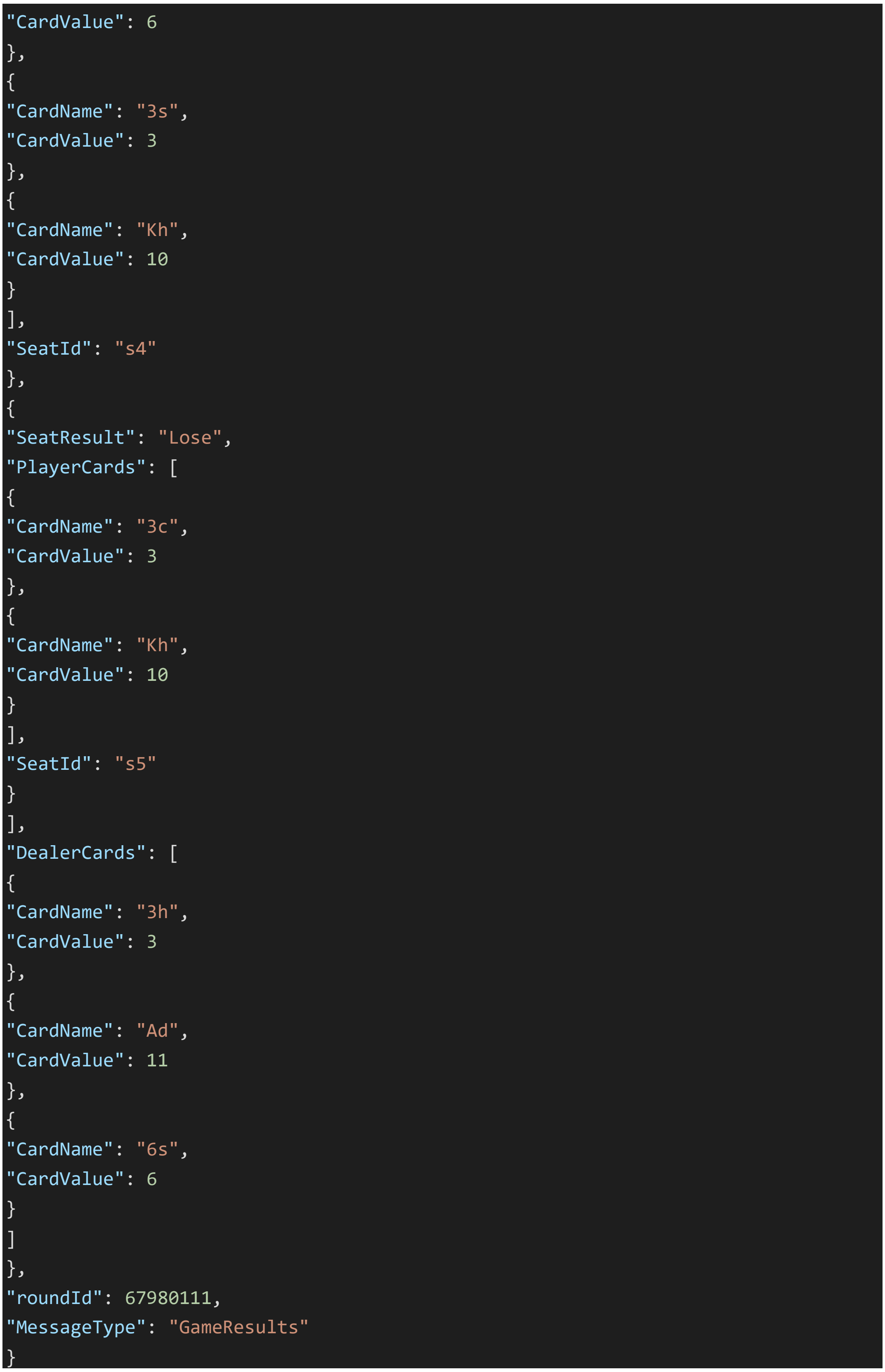
```
{
 "gameType": ,
 "destination": "gameNotifications",
 "TableId": "",
 "GameResults": {
 },
 "LastWinners": [
 {
 "Nickname": "",
 "winningAmount":
 },
 ],
 "roundId": ,
 "MessageType": "GameResults"
}
```

The message delivers the clients last round results, including the last winners list.

Examples:

5.8.1 Blackjack:

```
{
"TableId": "1",
"gameType": 1,
"LastWinners": [
{
"OperatorID": "11000321",
"WinAmount": 10000.0,
"Nickname": "player1",
"SeatId": "s4"
},
{
"OperatorID": "11000322",
"WinAmount": 0.0,
"Nickname": "pearl967",
"SeatId": "s5"
}
],
"destination": "gameNotifications",
"GameResults": {
"PlayerCards": [
{
"SeatResult": "Win",
"PlayerCards": [
{
"CardName": "2s",
"CardValue": 2
},
{
"CardName": "6c",
```



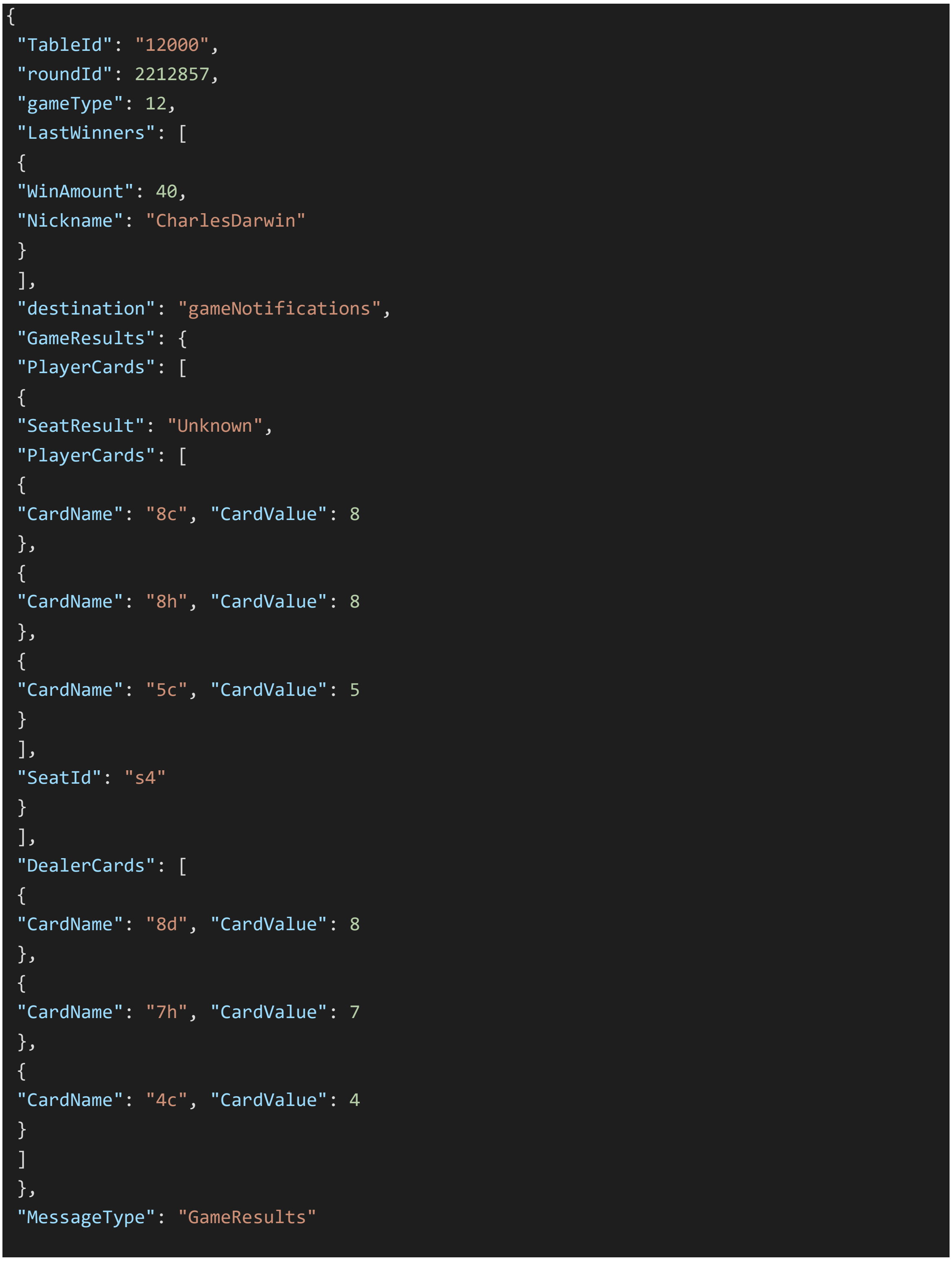
```
"CardValue": 6
},
{
"CardName": "3s",
"CardValue": 3
},
{
"CardName": "Kh",
"CardValue": 10
}
],
"SeatId": "s4"
},
{
"SeatResult": "Lose",
"PlayerCards": [
{
"CardName": "3c",
"CardValue": 3
},
{
"CardName": "Kh",
"CardValue": 10
}
],
"SeatId": "s5"
}
],
"DealerCards": [
{
"CardName": "3h",
"CardValue": 3
},
{
"CardName": "Ad",
"CardValue": 11
},
{
"CardName": "6s",
"CardValue": 6
}
]
},
"roundId": 67980111,
"MessageType": "GameResults"
}
```

5.8.2 Unlimited Blackjack:

***Important note:**

PlayerCards node contains all the cards dealt in the round (including cards marked as "Extra" for some players)

SeatResult node always Unknown (because can differ for different players)



```
{
 "TableId": "12000",
 "roundId": 2212857,
 "gameType": 12,
 "LastWinners": [
 {
 "WinAmount": 40,
 "Nickname": "CharlesDarwin"
 }
 ],
 "destination": "gameNotifications",
 "GameResults": {
 "PlayerCards": [
 {
 "SeatResult": "Unknown",
 "PlayerCards": [
 {
 "CardName": "8c", "CardValue": 8
 },
 {
 "CardName": "8h", "CardValue": 8
 },
 {
 "CardName": "5c", "CardValue": 5
 }
 ],
 "SeatId": "s4"
 }
 ],
 "DealerCards": [
 {
 "CardName": "8d", "CardValue": 8
 },
 {
 "CardName": "7h", "CardValue": 7
 },
 {
 "CardName": "4c", "CardValue": 4
 }
 ]
 },
 "MessageType": "GameResults"
```

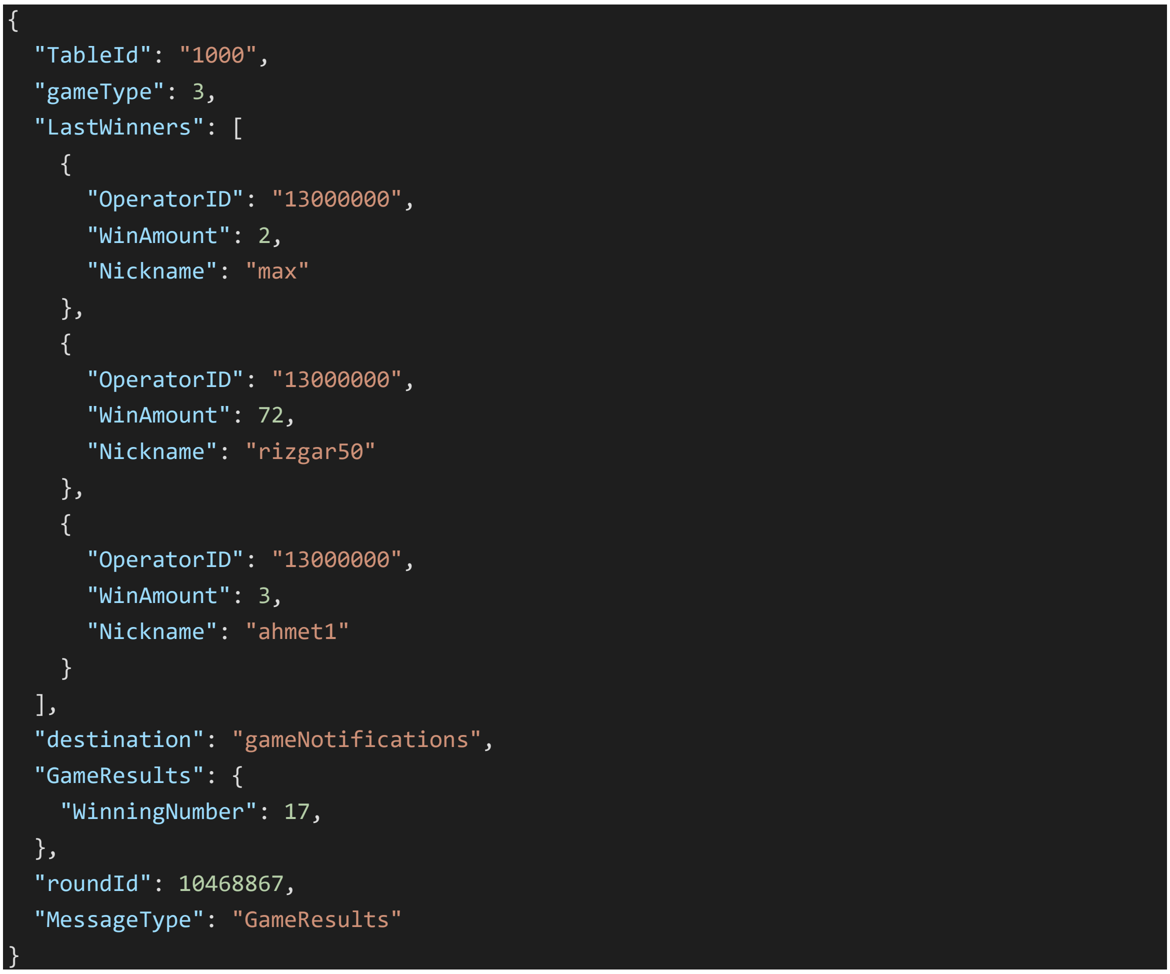
5.8.3 Baccarat:

```json
{
"TableId": "100",
"gameType": 2,
"destination": "gameNotifications",
"GameResults": {
"isPlayerPair": false,
"BankerCards": [
{
"CardName": "4c",
"CardValue": 4
},
{
"CardName": "Ts",
"CardValue": 10
}
],
"isBankerPair": false,
"isNaturalHand": true,
"PlayerCards": [
{
"CardName": "9s",
"CardValue": 9
},
{
"CardName": "Qc",
"CardValue": 10
}
],
"playerHandValue": 9,
"bankerHandValue": 4,
"roundId": 67977464,
"WinningHand": "Player",
"CardHandValue": 9
},
"roundId": 67977464,
"MessageType": "RoundEnded"
}
```

5.8.4 Dragon Tiger

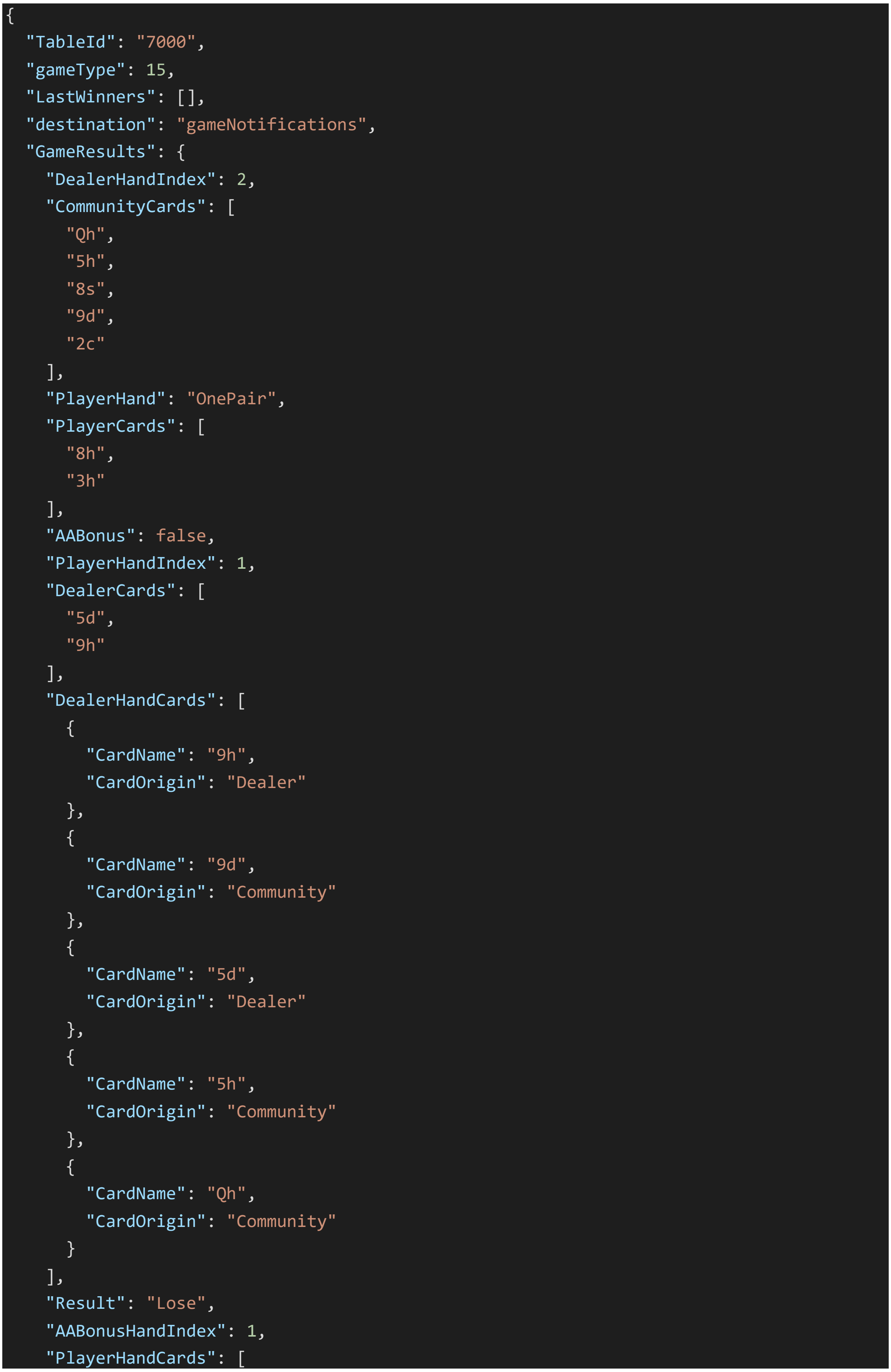
```
{
"TableId": "150",
"gameType": 24,
"destination": "gameNotifications",
"GameResults": {
"isPlayerPair": false,
"BankerCards": [
{
"CardName": "Ad",
"CardValue": 1
}
],
"isBankerPair": false,
"isNaturalHand": false,
"PlayerCards": [
{
"CardName": "Tc",
"CardValue": 10
}
],
"playerHandValue": 10,
"bankerHandValue": 1,
"roundId": 67977814,
"WinningHand": "Player",
"CardHandValue": 10
},
"roundId": 67977814,
"MessageType": "RoundEnded"
}
```

5.8.5 Roulette

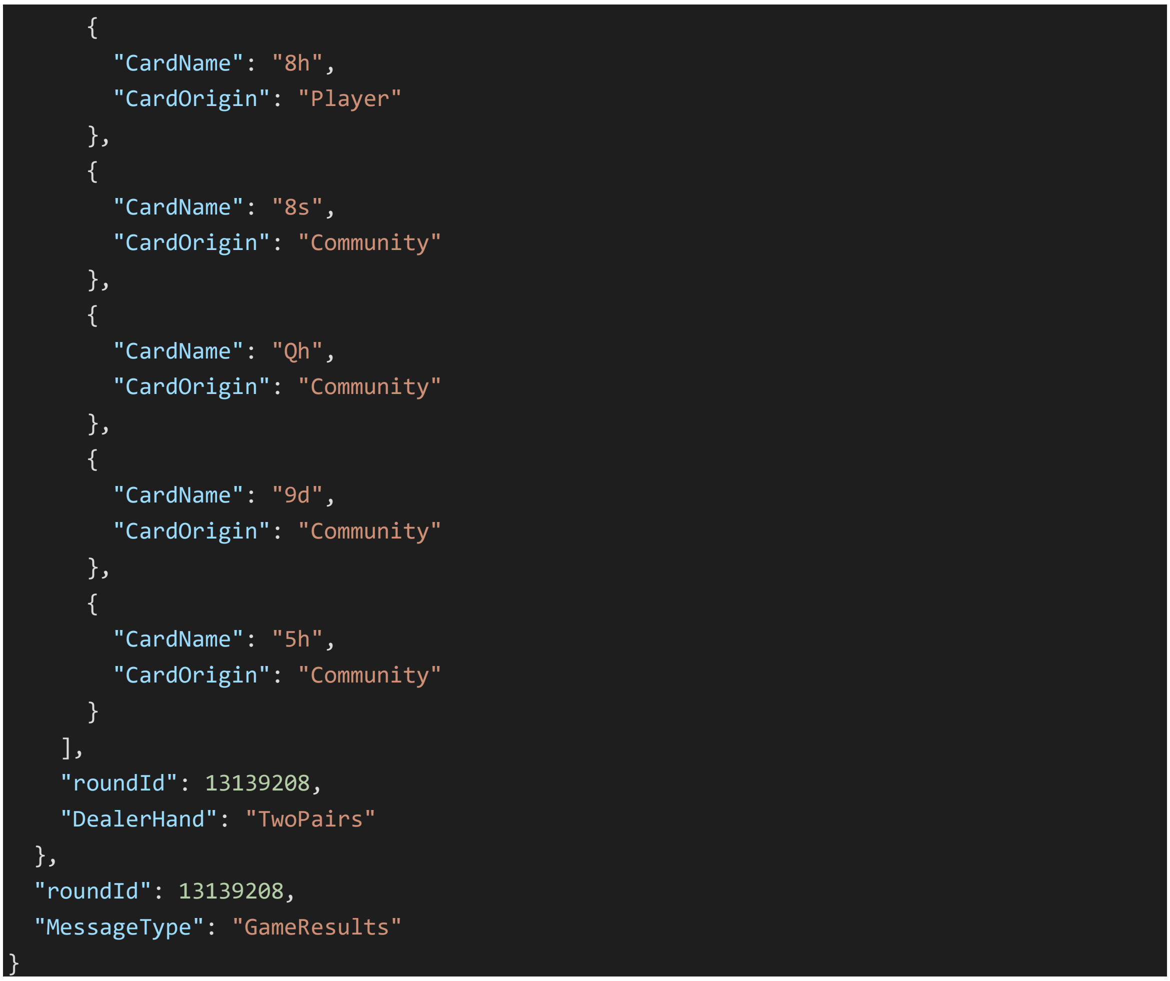


```
{
  "TableId": "1000",
  "gameType": 3,
  "LastWinners": [
    {
      "OperatorID": "13000000",
      "WinAmount": 2,
      "Nickname": "max"
    },
    {
      "OperatorID": "13000000",
      "WinAmount": 72,
      "Nickname": "rizgar50"
    },
    {
      "OperatorID": "13000000",
      "WinAmount": 3,
      "Nickname": "ahmet1"
    }
  ],
  "destination": "gameNotifications",
  "GameResults": {
    "WinningNumber": 17,
  },
  "roundId": 10468867,
  "MessageType": "GameResults"
}
```

5.8.6 Casino Hold’em

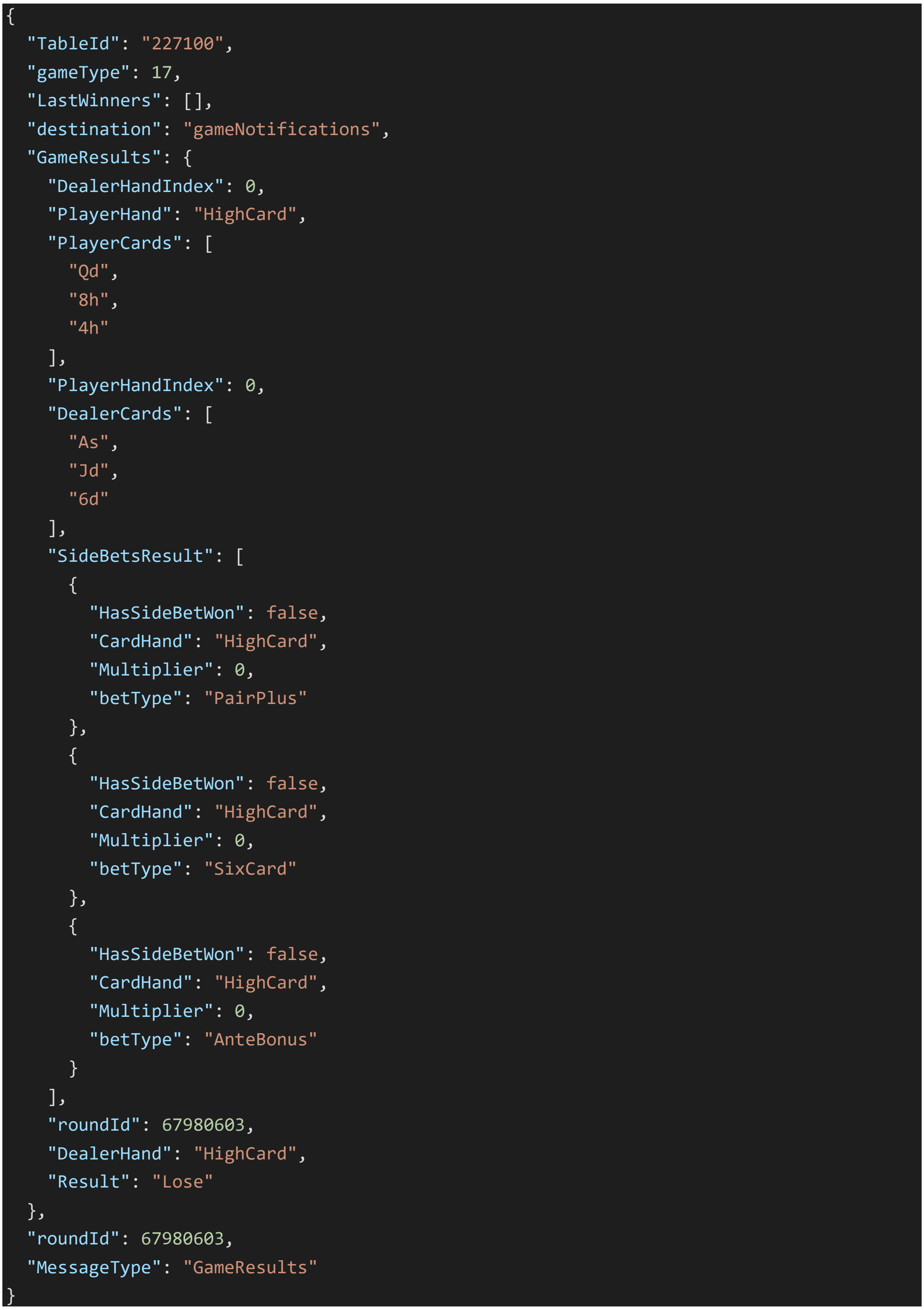


```
{
  "TableId": "7000",
  "gameType": 15,
  "LastWinners": [],
  "destination": "gameNotifications",
  "GameResults": {
    "DealerHandIndex": 2,
    "CommunityCards": [
      "Qh",
      "5h",
      "8s",
      "9d",
      "2c"
    ],
    "PlayerHand": "OnePair",
    "PlayerCards": [
      "8h",
      "3h"
    ],
    "AABonus": false,
    "PlayerHandIndex": 1,
    "DealerCards": [
      "5d",
      "9h"
    ],
    "DealerHandCards": [
      {
        "CardName": "9h",
        "CardOrigin": "Dealer"
      },
      {
        "CardName": "9d",
        "CardOrigin": "Community"
      },
      {
        "CardName": "5d",
        "CardOrigin": "Dealer"
      },
      {
        "CardName": "5h",
        "CardOrigin": "Community"
      },
      {
        "CardName": "Qh",
        "CardOrigin": "Community"
      }
    ],
    "Result": "Lose",
    "AABonusHandIndex": 1,
    "PlayerHandCards": [
```



```
      {
        "CardName": "8h",
        "CardOrigin": "Player"
      },
      {
        "CardName": "8s",
        "CardOrigin": "Community"
      },
      {
        "CardName": "Qh",
        "CardOrigin": "Community"
      },
      {
        "CardName": "9d",
        "CardOrigin": "Community"
      },
      {
        "CardName": "5h",
        "CardOrigin": "Community"
      }
    ],
    "roundId": 13139208,
    "DealerHand": "TwoPairs"
  },
  "roundId": 13139208,
  "MessageType": "GameResults"
}
```

5.8.7 Three Card Poker

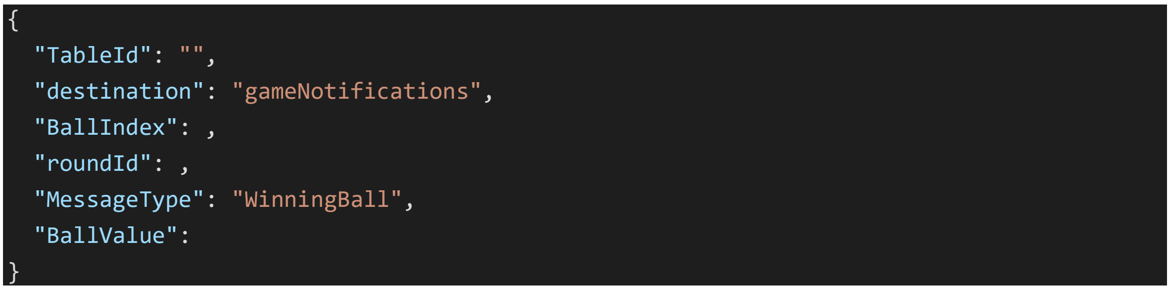


```
{
  "TableId": "227100",
  "gameType": 17,
  "LastWinners": [],
  "destination": "gameNotifications",
  "GameResults": {
    "DealerHandIndex": 0,
    "PlayerHand": "HighCard",
    "PlayerCards": [
      "Qd",
      "8h",
      "4h"
    ],
    "PlayerHandIndex": 0,
    "DealerCards": [
      "As",
      "Jd",
      "6d"
    ],
    "SideBetsResult": [
      {
        "HasSideBetWon": false,
        "CardHand": "HighCard",
        "Multiplier": 0,
        "betType": "PairPlus"
      },
      {
        "HasSideBetWon": false,
        "CardHand": "HighCard",
        "Multiplier": 0,
        "betType": "SixCard"
      },
      {
        "HasSideBetWon": false,
        "CardHand": "HighCard",
        "Multiplier": 0,
        "betType": "AnteBonus"
      }
    ],
    "roundId": 67980603,
    "DealerHand": "HighCard",
    "Result": "Lose"
  },
  "roundId": 67980603,
  "MessageType": "GameResults"
}
```

5.8.8 Andar Bahar

```
{
  "TableId": "228000",
  "gameType": 38,
  "destination": "gameNotifications",
  "GameResults": {
    "BaharCards": [
      "8c",
      "4d",
      "8h",
      "Qs",
      "Ac",
      "Td",
      "3d",
      "Jc",
      "8s",
      "Jh",
      "2s",
      "6c",
      "2h",
      "9d"
    ],
    "RoundId": 67978017,
    "JokerCard": "Kd",
    "WinningHand": "Andar",
    "AndarCards": [
      "Ad",
      "7s",
      "2c",
      "Th",
      "4s",
      "7d",
      "5c",
      "Qc",
      "Ts",
      "3s",
      "Qh",
      "5h",
      "Js",
      "3h",
      "Kh"
    ]
  },
  "roundId": 67978017,
  "MessageType": "RoundEnded"
}
```

5.9 Winning Ball



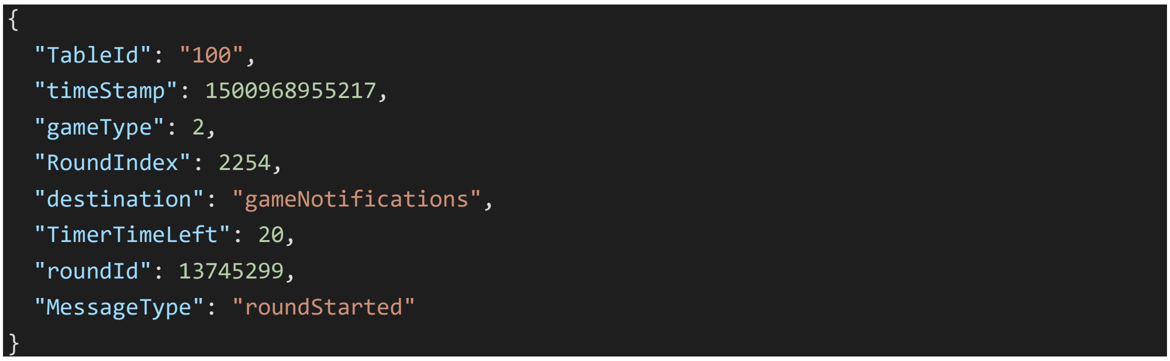
```
{
  "TableId": "",
  "destination": "gameNotifications",
  "BallIndex": ,
  "roundId": ,
  "MessageType": "WinningBall",
  "BallValue":
}
```

5.10 Round Started:

The message notifies the client that a round has started on the table.

The message will indicate the clients that the bets are open on the table and will deliver the time left till the bets are closed. *Timestamp is sent in UTC msec

Example:

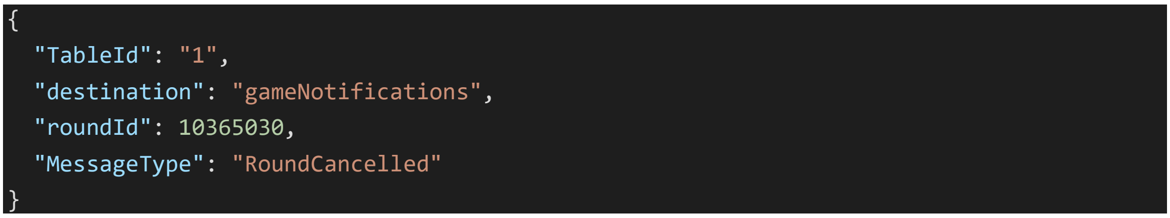


```
{
  "TableId": "100",
  "timeStamp": 1500968955217,
  "gameType": 2,
  "RoundIndex": 2254,
  "destination": "gameNotifications",
  "TimerTimeLeft": 20,
  "roundId": 13745299,
  "MessageType": "roundStarted"
}
```

5.11 Round Cancelled:

The message will notify the clients on round that was cancelled.

Message:



```
{
  "TableId": "1",
  "destination": "gameNotifications",
  "roundId": 10365030,
  "MessageType": "RoundCancelled"
}
```

5.12 Error Messages:

Message:

```
{
  "ErrorCode": ,
  "ErrorDescription": "",
  "MessageType": "Error"
}
```

The message delivers the client an error code and an error description.

The message will be sent if there was an error with the request that was sent from the client to the server.

Examples:

5.12.1 Wrong Table Id:

```
{
  "ErrorCode": 100,
  "ErrorDescription": "Wrong table id",
  "MessageType": "Error"
}
```

5.12.2 Table Inactive:

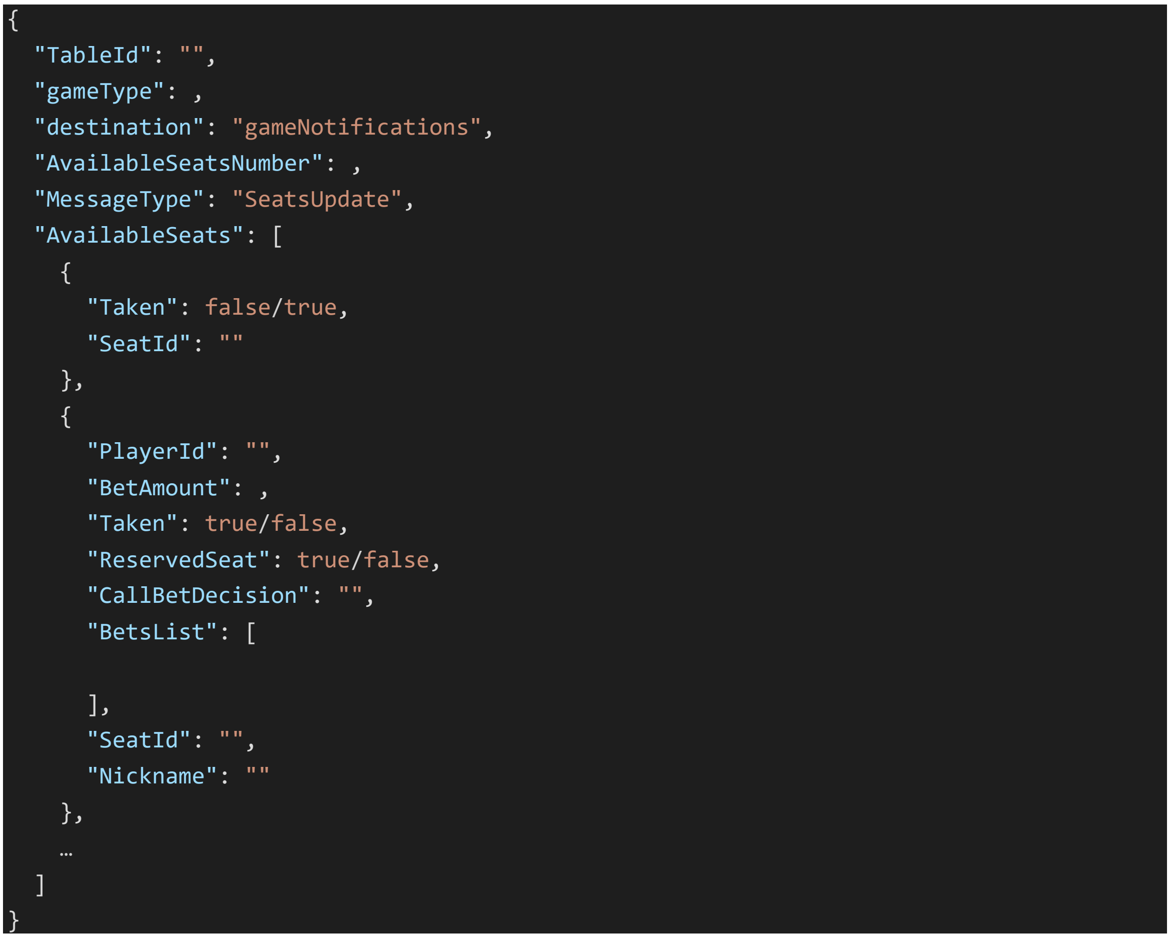
```
{
 "ErrorDescription": "Table inactive",
 "ErrorCode": 102,
 "MessageType": "Error"
}
```

5.12.3 Maximum allowed sessions number exceeded

```
{
  "ErrorDescription": "Maximum allowed sessions number exceeded",
  "ErrorCode": 30057,
  "MessageType": "Error"
}
```

5.12.4 Seats Update (Only games that have table seats):

Message:



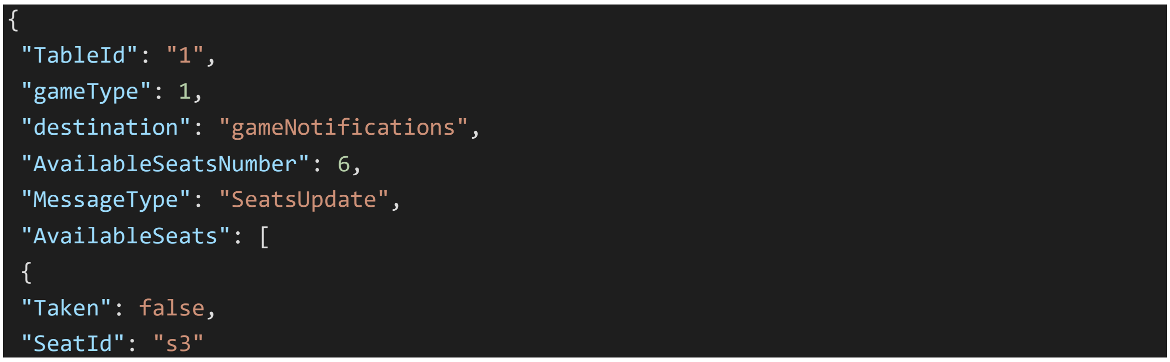
```
{
  "TableId": "",
  "gameType": ,
  "destination": "gameNotifications",
  "AvailableSeatsNumber": ,
  "MessageType": "SeatsUpdate",
  "AvailableSeats": [
    {
      "Taken": false/true,
      "SeatId": ""
    },
    {
      "PlayerId": "",
      "BetAmount": ,
      "Taken": true/false,
      "ReservedSeat": true/false,
      "CallBetDecision": "",
      "BetsList": [

      ],
      "SeatId": "",
      "Nickname": ""
    },
    …
  ]
}
```

The message delivers the status of the seats on a Black Jack table. This message will be sent id the status was changes (Seat taken or released). The message arrives if operator is registered by table id.

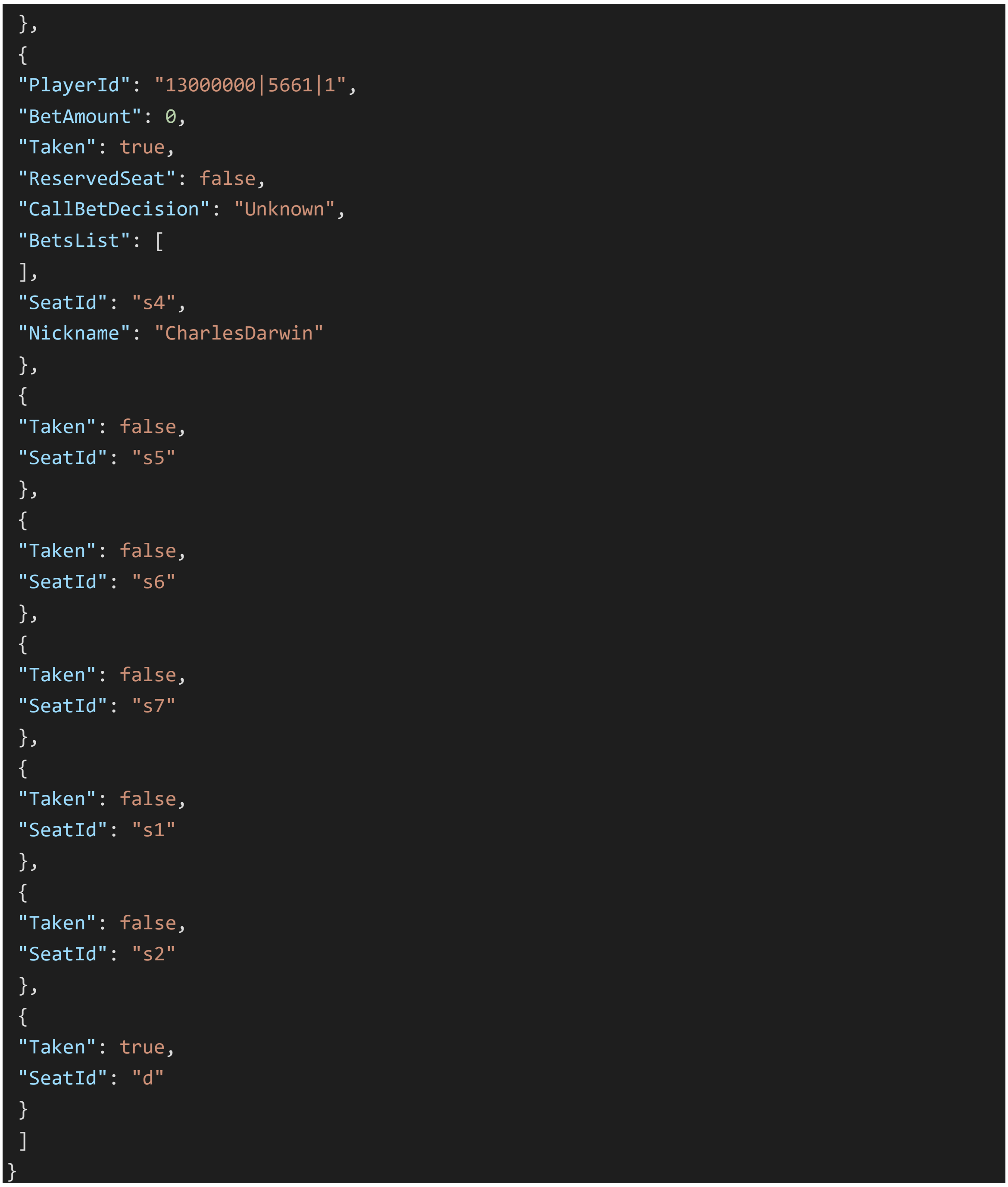
***Unlimited BJ remark**

Unlimited BJ doesn't send the message (although internally uses seats) because it has no sense for that kind of BJ game

Example:



```
{
 "TableId": "1",
 "gameType": 1,
 "destination": "gameNotifications",
 "AvailableSeatsNumber": 6,
 "MessageType": "SeatsUpdate",
 "AvailableSeats": [
 {
 "Taken": false,
 "SeatId": "s3"
```



```
},
{
"PlayerId": "13000000|5661|1",
"BetAmount": 0,
"Taken": true,
"ReservedSeat": false,
"CallBetDecision": "Unknown",
"BetsList": [
],
"SeatId": "s4",
"Nickname": "CharlesDarwin"
},
{
"Taken": false,
"SeatId": "s5"
},
{
"Taken": false,
"SeatId": "s6"
},
{
"Taken": false,
"SeatId": "s7"
},
{
"Taken": false,
"SeatId": "s1"
},
{
"Taken": false,
"SeatId": "s2"
},
{
"Taken": true,
"SeatId": "d"
}
]
}
```

5.13 Seats Status (Only games that have table seats):

Message:

```
{
 "TableId": "",
"gameType": ,
 "destination": "gameNotifications",
 "sessionId": "",
 "MessageType": "SeatsStatus",
 "Seats": [
 {
 "PlayerOnSeat": "",
 "Taken": true/false,
 "SeatId": ""
 },
 …
 ]
}
```

The message will be sent from the client on order to request seats status of the results from a given table. The server will reply with a SeatsStatus message.

***Unlimited BJ remark**

Unlimited BJ doesn't send the message (although internally uses seats) because it has no sense for that kind of BJ game

Example:

```
{
 "TableId": "1",
 "gameType": 1,
 "destination": "gameNotifications",
 "MessageType": "SeatsStatus",
 "Seats": [
 {
 "Taken": false,
 "SeatId": "s3"
 },
 {
 "Taken": false,
 "SeatId": "s4"
 },
 {
 "Taken": false,
 "SeatId": "s5"
 },
 {
 "Taken": false,
 "SeatId": "s6"
 },
 {
 "Taken": false,
 "SeatId": "s7"
 },
 {
 "PlayerOnSeat": "Martin Luther King",
 "Taken": true,
 "SeatId": "s1"
 },
 {
 "Taken": false,
 "SeatId": "s2"
 }
 ]
}
```

5.14 Dealer Changed:

The message delivers the name and the link to the picture of the new dealer on the table once a dealer was changed.

Example:

```
{
  "TableId": "1",
  "PictureLink":
"/LiveCasinoNew/BO/BizConfigs/Casino/Resources/Dealers/Images/Cristina_16.png",
  "destination": "gameNotifications",
  "DealerId": "78",
  "DealerName": "Cristina",
  "MessageType": "DealerChanged"
}
```

5.15 Shuffle:

The message will signal all of the registered sessions that a shuffle has occurred on the table.

Example:

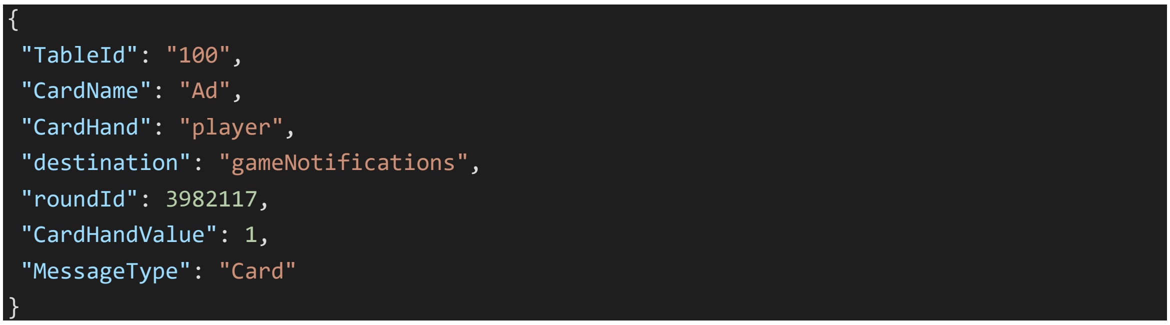
```
{
  "TableId": "5000",
  "gameType": 9,
  "destination": "gameNotifications",
  "MessageType": "Shuffle"
}
```

5.16 Game specific Server to Client Messages

5.16.1 Baccarat Card Message

The message will deliver the clients the card details: card value, card name and card hand.

Example:

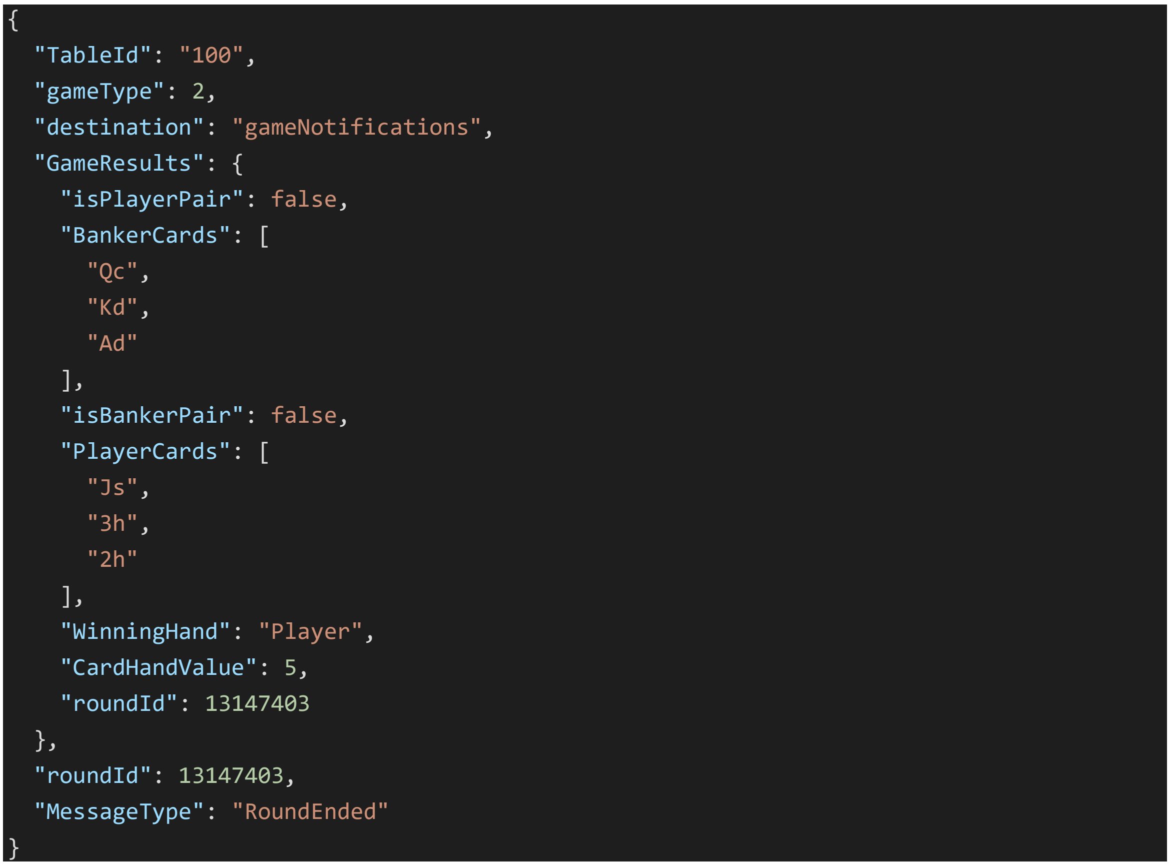


```
{
 "TableId": "100",
 "CardName": "Ad",
 "CardHand": "player",
 "destination": "gameNotifications",
 "roundId": 3982117,
 "CardHandValue": 1,
 "MessageType": "Card"
}
```

5.16.2 End Round message

The message notifies the client that round is ended.

Example:



```
{
  "TableId": "100",
  "gameType": 2,
  "destination": "gameNotifications",
  "GameResults": {
    "isPlayerPair": false,
    "BankerCards": [
      "Qc",
      "Kd",
      "Ad"
    ],
    "isBankerPair": false,
    "PlayerCards": [
      "Js",
      "3h",
      "2h"
    ],
    "WinningHand": "Player",
    "CardHandValue": 5,
    "roundId": 13147403
  },
  "roundId": 13147403,
  "MessageType": "RoundEnded"
}
```

5.16.3 Baccarat Round Started:

The message notifies the client that a round has started on the table.

The message will indicate the clients that the bets are open on the table and will deliver the time left till the bets are closed.

*For a Baccarat tables only, an additional parameter RoundIndex has been added. This parameter shows number of round from the last shuffle.

Example:

```
{
  "TableId": "100",
  "gameType": 2,
  "RoundIndex": 15,
  "destination": "gameNotifications",
  "TimerTimeLeft": 20,
  "roundId": 13147291,
  "MessageType": "roundStarted"
}
```

5.16.4 Card Hands Message

Delivers card hands in their original order with each inserted card.

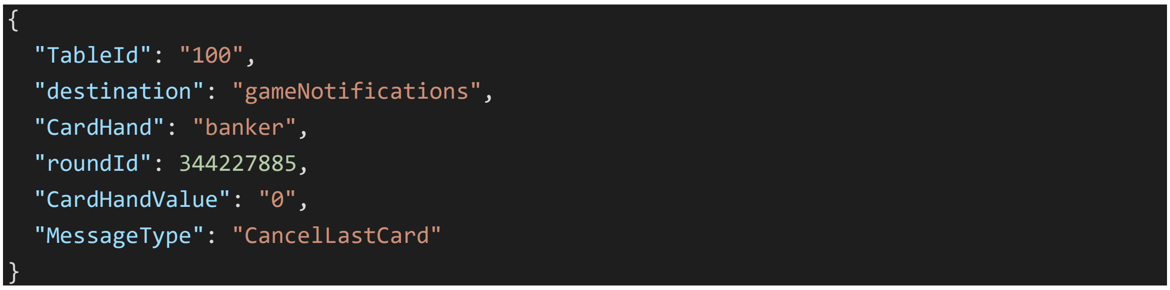
Example:

```
{
  "TableId": "667100",
  "BankerCards": [
    "Jh",
    "9d"
  ],
  "destination": "gameNotifications",
  "PlayerCards": [
    "7d",
    "6c"
  ],
  "roundId": 3966235,
  "MessageType": "CardHands"
}
```

5.16.5 Cancel Last Card Message

The message delivers the last cancelled card details: card hand, card hand new value and round Id once the "cancel card" event has occurred.

Example:



```
{
  "TableId": "100",
  "destination": "gameNotifications",
  "CardHand": "banker",
  "roundId": 344227885,
  "CardHandValue": "0",
  "MessageType": "CancelLastCard"
}
```

6. Dictionaries and Naming Conventions

6.1 Game Id codes

Game Id	Game Name
1	Blackjack
2	Baccarat
3	Roulette
4	Bet on Numbers
5	Hybrid Blackjack
6	Keno
7	Automatic Roulette
8	Wheel of Dice
9	Sede
10	American Blackjack
11	American Hybrid Blackjack
12	Unlimited Blackjack
14	Sic BO
15	Casino Hold'em
17	Three Card Poker(NJ)/ Teen Patti
20	Baccarat KO
21	Baccarat Super 6
24	Dragon Tiger
25	No Commission Baccarat
26	Baccarat Dragon Bonus
27	BaccaratQueenco
28	BaccaratPuntoBanco
29	RoulettePortomaso
31	American Roulette
32	Triple Roulette
38	Andar Bahar

6.2 Card Values

Code	Card Value
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
T	10
J	Jack
Q	Queen
K	King
A	Ace

6.3 Card Suits

Code	Card Suit
d	Diamonds
h	Hearts
s	Spades
c	Clubs

6.4 Token PHP; Code Example

To be able to work with 3rd party API you must implement token authentication, see API Usage:

```php
<?php if($_REQUEST['read'] == "yes")
{          echo
file_get_contents('/tmp/token.txt');
        return;
}
$json_obj = json_decode(file_get_contents('php://input'));
file_put_contents('/tmp/token.txt', $json_obj -> Token);
```