

Ezugi Live Casino



Smart Move

Seamless Wallet API

Document History

Version	Description	Author	Date
v3.1.2	<ol style="list-style-type: none"> P 3.1.2 BetType Ids has been updated P 7 Integration into LC Staging Environment has been updated with a new launch URL P 8.2.3 Examples Per Game has been updated with Jackpot request examples 	Marianna Reva	March 2019
v3.1.3	<ol style="list-style-type: none"> P 6.1.2 Game Ids Codes have been updated. A new gameld for Andar Bahar has been added 	Marianna Reva	June 2019
v3.1.4	<ol style="list-style-type: none"> New games Three Card Poker and No Commission Baccarat have been added to: <ol style="list-style-type: none"> P.5.1.2 Launch Url, open table list P 6.1.2 Game Id Codes P.6.1.3 Bet type Ids codes 	Marianna Reva	August 2019
v3.1.5	<ol style="list-style-type: none"> New games Three Card Port and No Commission Baccarat have been added to: <ol style="list-style-type: none"> P.5.1.2 Launch Url, open table list P 6.1.2 Game Id Codes P 6.1.3 	Marianna Reva	March 2020
v3.1.6	<ol style="list-style-type: none"> API Retransmission Policy & Error Handling of Debit call has been updated New Game Type has been added to the p. 6.1.2 game ids P. 5.1.3 Launch URL has been updated, see openTable params Jackpot Roulette game is terminated, and the game example has been removed from the p.8.2 examples Bet Behind has been removed from the examples. 	Marianna Reva	May 2021
v3.1.7	<ol style="list-style-type: none"> Edited typos API Retransmission Policy & Error Handling of Debit call has been updated Updated description for input and output parameters Updated gameld's list with upcoming games (6.1.2) Edited the requests from 8. Appendix 	Daniel Dinulescu, Razvan Andrei	October 2021
v3.1.8	<ol style="list-style-type: none"> Reformatted document Removed pingUrl parameter Added new Gameld's Updated Bet Codes and Error Codes info Updated Request/Responses section 	Razvan Andrei, Andrei Gheorghisor Ning Hsu	Nov 2022

Contents

1. Document Overview	5
2. Integration Overview	5
3. Protocol.....	5
4. Security	5
5. Integration API's.....	7
5.1 Open Game / Lobby	7
5.1.1 Description.....	7
5.1.2 Launch URL Parameters	7
5.2 Authentication of User API.....	9
5.2.1 Description.....	9
5.2.2 HTTP Header	9
5.2.3 Input Parameters	9
5.2.4 Output Parameters	9
5.2.5 API Retransmission Policy & Error Handling	11
5.3 Debit API.....	12
5.3.1 Description.....	12
5.3.2 HTTP Header	12
5.3.3 Input Parameters	13
5.3.4 Output Parameters	14
5.3.5 API Retransmission Policy & Error Handling	14
5.4 Rollback API.....	15
5.4.1 Description.....	15
5.4.2 HTTP Header	16
5.4.3 Input Parameters	17
5.4.4 Output Parameters	18
5.4.5 API Retransmission Policy & Error Handling	19
5.5 Credit API.....	20
5.5.1 Description.....	20
5.5.2 HTTP Header	20
5.5.3 Input Parameters	21
5.5.4 Output Parameters	23
5.5.5 API Retransmission Policy & Error Handling	24
5.6 Get New Token	25
5.6.1 Description.....	25

5.6.2 Input Parameters	25
5.6.3 Input Parameters	25
5.6.4 Output Parameters	26
5.6.5 API Retransmission Policy & Error Handling	26
6. Dictionaries and Naming Convention	27
6.1 Dictionaries.....	27
6.1.1 Return reason codes	27
6.1.2 Game ID codes	27
6.1.3 Bet Type codes.....	29
6.1.4 Error Codes	30
6.1.5 Languages Dictionary (iso 639-1).....	30
6.1.6 Currencies Dictionary (iso 4217)	31
6.2 Naming Conventions	31
6.2.1 Field Key Names for All API Calls.....	31
7. Integration into Live Casino Staging Environment.....	32
7.1 Integration Server Details:.....	32
7.2 Launching Lobby and Games:	32
7.3 Game Size and Ratio:.....	32
8. Appendix	33
8.1 Production Studios Information	33
8.2 Examples	33
8.2.1 Authentication Example	33
8.2.2 Get New Token Example.....	33
8.3 Examples Per Game.....	34
8.3.1 Blackjack	34
8.3.2 Baccarat	36
8.3.3 Casino Hold'em	38
8.3.4 Tip example.....	42
8.3.5 GameDataString Example	43
8.3.6 Rollback Example	44
8.3.7 Hash Signature Examples.....	45
8.3.8 Error Code Examples.....	46

1. Document Overview

This document describes an integrated system which enables the Live Casino backend to integrate with online operators. It includes the financial transactions, login calls and launch protocols used.

2. Integration Overview

The Live Casino is a B2B system which allows shops and online operators to offer games to their clients through their existing systems. A simple call to “open game” will generate the casino system start up by summoning the operator to open a customized lobby. Once open, the user may navigate freely throughout the Live Casino and play any game while tracking his balance. Four simple API calls allow the casino to credit and debit the player’s account in real time using seamless wallet integration. All the financial calls are monitored and managed by the casino system to ensure no financial transaction is lost.

3. Protocol

The integration uses HTTP’s POST communications based on JSON message type. The operator must use HTTPS to communicate with our server, but we need to configure it in advance. All the field types specified in this document refer to Java and MySQL field type definitions. All responses should be with Status Code 200 for them to be read by the server since internal error codes will be used. Other status codes might trigger unexpected behavior.

4. Security

Connection (Mandatory):

- **SSL/TLS Communication:** All communication with the operator should be done over SSL secured channel. *In production, only HTTPS communication is permitted.*
- **IP Whitelisting:** The operator must whitelist the Ezugi server IPs on their servers and receive any kind of requests related to Ezugi from the registered IPs only. Any requests related to Ezugi that are being submitted that are not from the registered Ezugi IPs should be discarded and Ezugi should be notified.
- *Ezugi does not need to whitelist the operator’s IP except for BO API, Push Balance API, and restricted markets.* Contact the Integration manager or KAM for accessing these APIs.
- **Hash Signature Check:** Operators will receive a hash signature on all requests coming from Ezugi signed by a secret key that will be issued by Ezugi specifically for the operator. The hash signature will be created by using SHA256 hash algorithm, the output format is base64 encoding. The operator should compare the received hash signature (*in the request header*) with their own signature that will be created by using the secret key and the received request.

Token (Mandatory):

- **New Token Creation on Authentication:** On each authentication for a specific player, our system will need to receive a new generated token. No token reuse will be accepted.
- **Token Expiration:** Any token (initial authentication or session) issued by an operator should have an expiration date and should expire if not used after a short period of time. Initial authentication tokens should expire after 30-60 seconds, session (financial) tokens, should expire after 30-40 minutes of inactivity. Shorter sessions than that are possible, but no longer than 40 minutes.

Round Closing Options (Highly recommended):

- **Round Closure:** Every debit will always be followed by a credit (or rollback) and the parity between them can be used to close the round. Our transactionId's format will always mark the debit with a "d" respectively a credit with a "c", keeping the rest of the UUID identical, for an easier correlation between the two.
 - (Optional) An operator will receive a credit request carrying a round closure notification (**endRound=true**). This credit request will mark the end of the round for each player. No other game credit request carrying the same round ID should be processed by the operator.
 - Exception for the above is Blackjack, where each hand (max 2 hands per seat) will have a transaction closing the round for the respective hand. Operator can consider the round being closed after all hands are closed, "creditIndex" can also be used for this purpose.
 - If the number of credits is different to that of debit requests, please notify the Integration Manager
- **Corresponding Request Check:** When the operator receives a credit request, a check must be performed that a corresponding debit request was made carrying the same tableId, gameId, roundId, debitTransactionId, seatId (seat Id is relevant for Blackjack only).

Important notes!

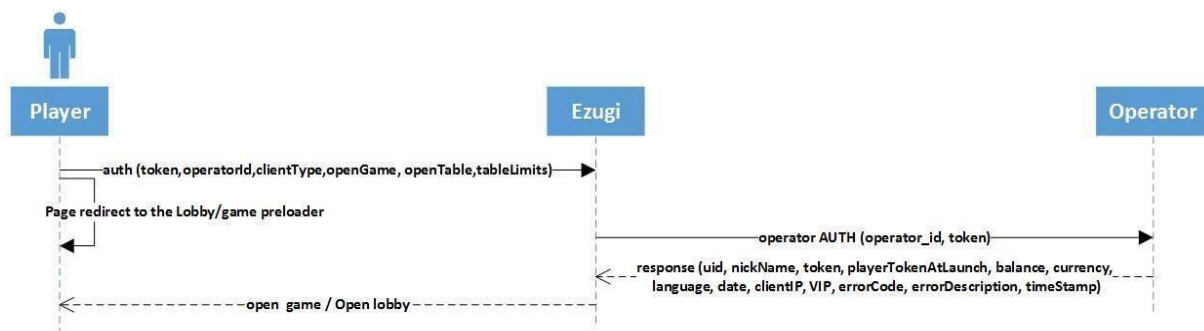
- Rounds should be closed individually per player by your system
- In seat-based games such as Blackjack and Unlimited Blackjack, each seat will be closed individually.

5. Integration API's

5.1 Open Game / Lobby

5.1.1 Description

The following sequence diagram describes the communication between the Player, Ezugi and the operator for every call to open the lobby or games.



The following table describes the parameters supplied by the operator to the Live Casino in order to launch the game lobby.

5.1.2 Launch URL Parameters

Parameter Name	Data Type/ Description	Optional/Mandatory
operatorId	Int (11)	Mandatory
	Defines the particular operator that the player belongs to. This is allocated by the Live Casino system in advance.	
token	String / VARCHAR (250)	Mandatory
	Initial token generated by the operator based upon the username and password. Will be used "as is" in order to authenticate the user. A recommended token format for all of our operators is UUID. A token should have at least 20 symbols which should be a mix of digits and characters. Any token (initial authentication or financial) issued by an operator should have an expiration date and should expire if not used after a short period of time. Initial authentication tokens should expire after 30-60 seconds.	
language	CHAR (2)	Optional
	Specifies the desired language file to load for the current game. Default English (en).	
clientType	VARCHAR	Optional
	html5	
selectGame	VARCHAR (30)	Optional
	Will open the lobby to the tab of the selected game. The possible values are: "blackjack", "roulette", "poker", "baccarat", "dragontiger_sicbo", "lottery" For example, "selectGame=blackjack" will open the lobby to the blackjack tab showing only the blackjack games. A player can then navigate to other games through the lobby. <i>Depending on the market, the selectGame values might differ</i>	
openTable	Int (11)	Optional
	Appending the tableId will open a specific table of a certain game. The possible values for integration server are: -Blackjack – 1 -Unlimited BJ – 51 -Roulette – 1000	

	<ul style="list-style-type: none"> -Auto Roulette – 5001 -OTT Roulette – 611000 -EZ Dealer Roulette – 481001 -Casino Hold'em – 507000 -Teen Patti – 227100 -Bet on Teen Patti – 227101 -One Day Teen Patti Classic – 227103 -Royal Poker – 537000 -Andar Bahar – 228000 -OTT Andar Bahar – 328000 -Ultimate Andar Bahar – 228100 -Lucky 7 – 228001 -32 Cards – 228002 -Baccarat – 100 -OTT Baccarat – 30100 -Baccarat KO – 120 -Baccarat Super 6 – 130 -Dragon Tiger – 150 -No Commission Baccarat – 170 -Lottery – 602000 -Sic-Bo – 204000 -American BJ – 10 (New Jersey only) -American Roulette – 1010 (New Jersey only) -Baccarat Dragon Bonus – 140 (New Jersey only) -Three Card Poker – 7100 (New Jersey only) <p>*Production table values are provided in the Ezugi Online Production Game Details document.</p>	
tableLimits	Int (10)	Optional
	To determine the limit for the new table being opened, add a limit ID in the URL (e.g. tableLimits=1)	
homeUrl	Encoded URL	Optional
	<p>Sending this parameter in URL means on EVERY home click the user will be redirected to this url (from lobby, direct link game and game opened through the lobby in frame).</p> <p>The URL must contain HTTP/HTTPS prefix to work.</p> <p>The structure of the dynamic home Url is the following:</p> <p>homeUrl=[http/https][DOMAIN]?[KEY=VALUE]%26[KEY=VALUE]%26[KEY=VALUE]%26....</p>	
cashierUrl	Encoded URL	Optional
	<p>Dynamic cashier URL is a functionality to set cashier button url through url parameter: cashierUrl. Sending this parameter in URL means on EVERY cashier click the user will be redirected to this URL.</p> <p>Optional methods for opening cashier URL: popup/tab/post/replace.</p> <p>The URL must contain HTTP/HTTPS prefix to work.</p> <p>The structure of the dynamic home Url is the following:</p> <p>cashierUrl=[http/https] [DOMAIN]?[KEY=VALUE]%26[KEY=VALUE]%26[KEY=VALUE]%26.....</p>	
postMessage	Int	Optional
	<p>Value=1</p> <p>There is an option to use post message events within the lobby, to get the event from the lobby frame to its parent window (frame skin only).</p> <p>If the lobby is loaded as an iframe, the operator can choose to get a post message to the page containing the lobby. The parent page must implement a way to receive that message from the lobby iframe and set its own functionality.</p> <p>Post Message events can be configured separately, please get in contact with integration manager to properly configure it.</p>	

5.2 Authentication of User API

5.2.1 Description

The Authentication API enables the Live Casino system to verify the information received in the launch protocol. Only once the operator approves this information will the lobby load for the player.

5.2.2 HTTP Header

Parameter Name	Data Type/Description
hash	VARCHAR (100)
	Hash Signature Check: an operator will receive on all requests coming from Ezugi a hash signature signed by a secret key that will be issued by Ezugi specifically for the operator. The hash signature will be created by using SHA256 hash algorithm and will create a signature by using a secret key and the request that is sent to the operator. An output format is base64 encoding. The operator should compare the received hash signature with his own signature that will be created by using the secret key and the received request (the signature should be removed from the received request before calculating the signature on the operator's end).

5.2.3 Input Parameters

Parameter Name	Data Type/Description
operatorId	INT (11)
	Defines the particular operator that the player belongs to.
token	VARCHAR (250)
	Player's launch token that was previously used in the launch lobby/game URL by the operator launch the protocol. This is returned as the playerTokenAtLaunch in the operator response.
platformId	Int (10)
	Is an indication for the player's platform/device. Possible values are: 0 – HTML desktop, 2 – Smartphone, 3 – Tablet, 9 – Flash desktop, 1 – Mobile (*1- Mobile will be removed in the future)
timestamp	Long
	Time representation in UNIX milliseconds format. For example: "timestamp":1531141297495

5.2.4 Output Parameters

Parameter Name	Data Type/Description	Optional/Mandatory
operatorId	INT (11)	Mandatory
	The operator's unique identifier.	
uid (User ID)	VARCHAR (50)	Mandatory
	An operator's unique identification for each player. Please only use alphanumeric characters separated by hyphen, underscore, or dot	
nickName	VARCHAR (100)	Optional
	A player's nickname may include characters and numbers only. If no nickname is given to a player, a default generated nickname will appear in the lobby and blackjack game. <i>Warning: sending null, empty or special characters in the nickname will trigger errors.</i>	

token	VARCHAR (250)	Mandatory
	<p>Not to be confused with the launch token. This token is the session token; a unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session.</p> <p>A recommended token format for all our operators is UUID. A token should have at least 20 symbols which should be a mix of digits and characters. Any token (initial authentication or financial) issued by an operator should have an expiration date and should expire if not used after a short period of time.</p> <p>Financial tokens, if not used, should expire after 30-40 minutes.</p>	
playerTokenAtLaunch	VARCHAR (255)	Mandatory
	The player's initial token from operator (by launching the lobby). This token is equal to "&token=" used in <i>launchUrl</i>	
balance	Double (25,2)	Mandatory
	Available money in the player's wallet at the time of the request. The value will be presented in the currency of that session.	
currency	CHAR (4)	Mandatory
	The three-character currency code of the balance. Ratio can be specified by adding the character as a prefix (examples: USD, EUR, kVND)	
clientIP	VARCHAR (30)	Optional
	The IP of the user that is authenticating	
VIP	String	Optional
	The operator may specify a VIP level for each player to define the player's game limits. VIP values are numbers between "0" and "5"; "0" being the default value when no VIP is entered.	
errorCode	TINYINT (4)	Mandatory
	Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0.	
errorDescription	VARCHAR (100)	Mandatory
	A string that describes the response. This string is not the message to the player but rather gives details of the error.	
timestamp	VARCHAR (13)/Long	Mandatory
	<p>1. Time is represented in UNIX milliseconds format. An example and instructions: http://currentmillis.com/</p> <p>2. The Operator can return the timestamp of the balance with their own incremental counter rather than an actual timestamp.</p> <p>For example, "timestamp": "1", "timestamp": "2", "timestamp": "3", etc'</p> <p>In both cases the latest actual player's balance should come to the Ezugi system with the latest timestamp.</p>	

5.2.5 API Retransmission Policy & Error Handling

- **In the event of a final error response - The Live Casino server will NOT repeat the call.** A relevant message is presented to the user.
- In the event of:
 - **Client or Server Error** - Any potential error inside the operator's system excluding an HTTP exception type, such as 400, 402, 500 etc.
 - **Socket Timeout** - which may occur when the connection is closed by the system.
 - **Client's response with errorCode 10 (see p. 6.1.4. Error Codes)**
- **NOTE: All other error codes except 10 are considered final and the request will not be retried**

The Live Casino server will repeat the authentication call three times in ten second intervals (default) or according to the custom retry setting of the operator.

All operators are expected to gracefully handle any added or unknown parameters in the request data.

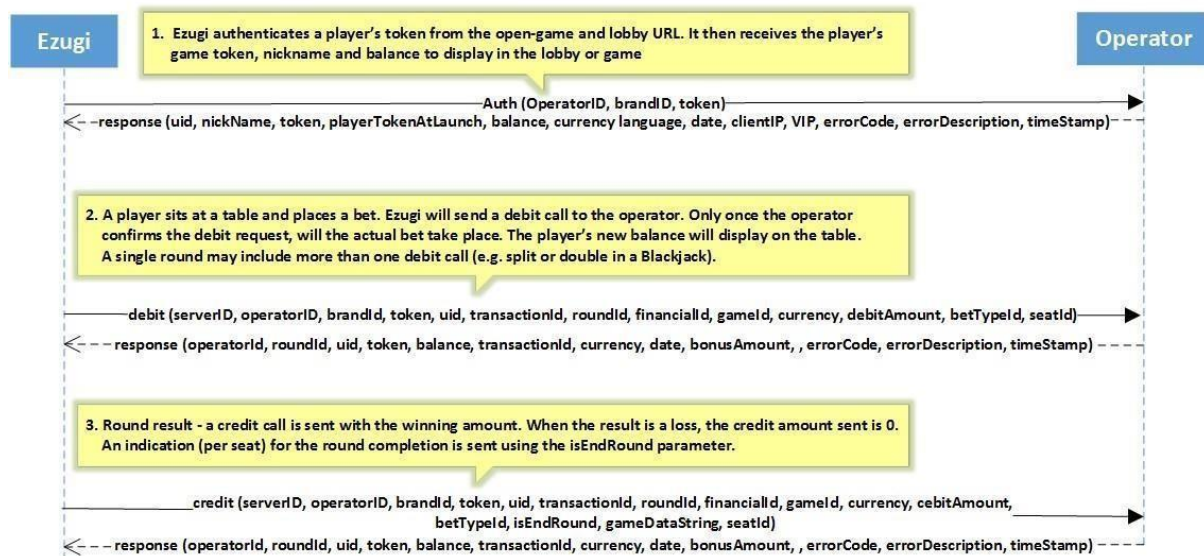
5.3 Debit API

5.3.1 Description

The Debit API enables the Live Casino platform to withdraw money from the player's wallet through the operator's site. This financial transaction will be done in three cases: When a regular bet is placed, a regular and a side bet are placed, and the dealer is tipped.

The following sequence diagram describes the communication that takes place between Ezugi and the operator for any standard debit or credit call.

Standard single debit and credit round communication example between Ezugi and the operator



5.3.2 HTTP Header

Parameter Name	Data Type/Description
hash	VARCHAR (100)
	<p>Hash Signature Check: an operator will receive on all requests coming from Ezugi a hash signature signed by a secret key that will be issued by Ezugi specifically for the operator. The hash signature will be created by using SHA256 hash algorithm and will create a signature by using a secret key and the request that is sent to the operator. The output format is base64 encoding. The operator should compare the received hash signature with his own signature that will be create by using the secret key and the received request</p> <p>(The signature should be removed from the received request before calculating the signature on the operator's end).</p>

5.3.3 Input Parameters

Parameter Name	Data Type/Description
serverId	INT (11)
	The unique serverId of the Live Casino system.
operatorId	INT (11)
	Defines the particular operator that the player belongs to.
token	VARCHAR (250)
	This token is the session token; a unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session. A recommended token format for all our operators is UUID. A token should have at least 20 symbols which should be a mix of digits and characters. Any token (initial authentication or financial) issued by an operator should have an expiration date and should expire if not used after a short period of time. Financial tokens, if not used, should expire after 30-40 minutes.
uid	VARCHAR (50)
	An operator's unique identification for each player. Please only use alphanumeric characters separated by hyphen, underscore, or dot.
transactionId	Char (50)
	A unique key to indicate a specific financial activity. This key will guarantee that a transaction is handled only once.
roundId	BigInt (18)
	A unique key per server ID, to identify a game cycle. Games with more than one financial cycle will have the same roundId for all requests.
gameId	TINYINT (4)
	See p.6.1.2. Game ID codes
tableId	INT (11)
	A unique key per table ID.
currency	Char (4)
	The currency code for the player's balance in a particular session.
debitAmount	Double (25,2)
	The amount requested to be withdrawn from the player's wallet. The value must be a positive number above zero.
betTypeID	TINYINT (4)
	The bet type code of player range refers to Appendix - 5.2 section
seatId	VARCHAR (10)
	The seat index number from where the debit occurred (Relevant only to BJ) "s1" to "s7" refers to the player's initial seat "s1-2" to "s7-2" refers to split seat
platformId	INT (10)
	Is an indication for the player's platform/device. Possible values are: 0 – HTML desktop, 2 – Smartphone, 3 – Tablet
timestamp	Long
	Time representation in UNIX milliseconds format. For example: "timestamp":1531141297495

5.3.4 Output Parameters

Parameter Name	Data Type/Description	Optional/Mandatory
operatorId	INT (11)	Mandatory
	The operator's unique identifier.	
roundId	BigInt (18)	Mandatory
	A unique key per server ID, to identify a game cycle. Games with more than one financial cycle will have the same roundId for all requests.	
uid	VARCHAR (50)	Mandatory
	A Player's unique identifier in the operator system. Please only use alphanumeric characters separated by hyphen, underscore, or dot	
token	VARCHAR (250)	Mandatory
	This token is the session token; a unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session. A recommended token format for all our operators is UUID. A token should have at least 20 symbols which should be a mix of digits and characters. Any token (initial authentication or financial) issued by an operator should have an expiration date and should expire if not used after a short period of time. Financial tokens, if not used, should expire after 30-40 minutes.	
balance	Double (25,2)	Mandatory
	Available money in the player's wallet at the time of the request. The value will be presented in the currency selected for the session	
transactionId	VARCHAR (50)	Mandatory
	The same transactionId as was sent in the request for this response. A unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once.	
currency	CHAR (4)	Mandatory
	The currency of the balance.	
bonusAmount	Double (25,2)	Optional
	The bonus amount from the total debit amount. * Cannot be more than debit amount.	
errorCode	TINYINT (4)	Mandatory
	Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0.	
errorDescription	VARCHAR (100)	Mandatory
	A string that describes the response. This string is not the message to the player but rather gives details of the error.	
timestamp	VARCHAR (13)/Long	Mandatory
	1. Time is represented in UNIX milliseconds format. An example and instructions: http://currentmillis.com/ 2. The Operator can return the timestamp of the balance with their own incremental counter rather than an actual timestamp. For example, "timestamp": "1", "timestamp": "2", "timestamp": "3", etc' In both cases the latest actual player's balance should come to the Ezugi system with the latest timestamp.	

5.3.5 API Retransmission Policy & Error Handling

In the event of a final error response or server error (status 400, 500) - The Live Casino server will NOT repeat the **debit** call.

All operators are expected to gracefully handle any added or unknown parameters in the request data.

5.4 Rollback API

5.4.1 Description

During a debit transaction, there may be some cases where the transaction should be cancelled, and therefore, rolled back. Rollback API is designed to deal with exceptions received from the operator (i.e., when the server disconnects, or an invalid error response is returned from the operator's end). The debit transaction is converted into a rollback transaction, **using the same transactionId as the debit**, and the debit amount field is renamed to rollback amount. It is crucial for the rollback to be in a separate method from the debit to avoid errors and data mix.

Rollback transactions are generated in response to the following events or errors received:

1. No response is received from the operator when no more bets are accepted.
2. Operator sends an exception.
3. Operator sends an error code 10.

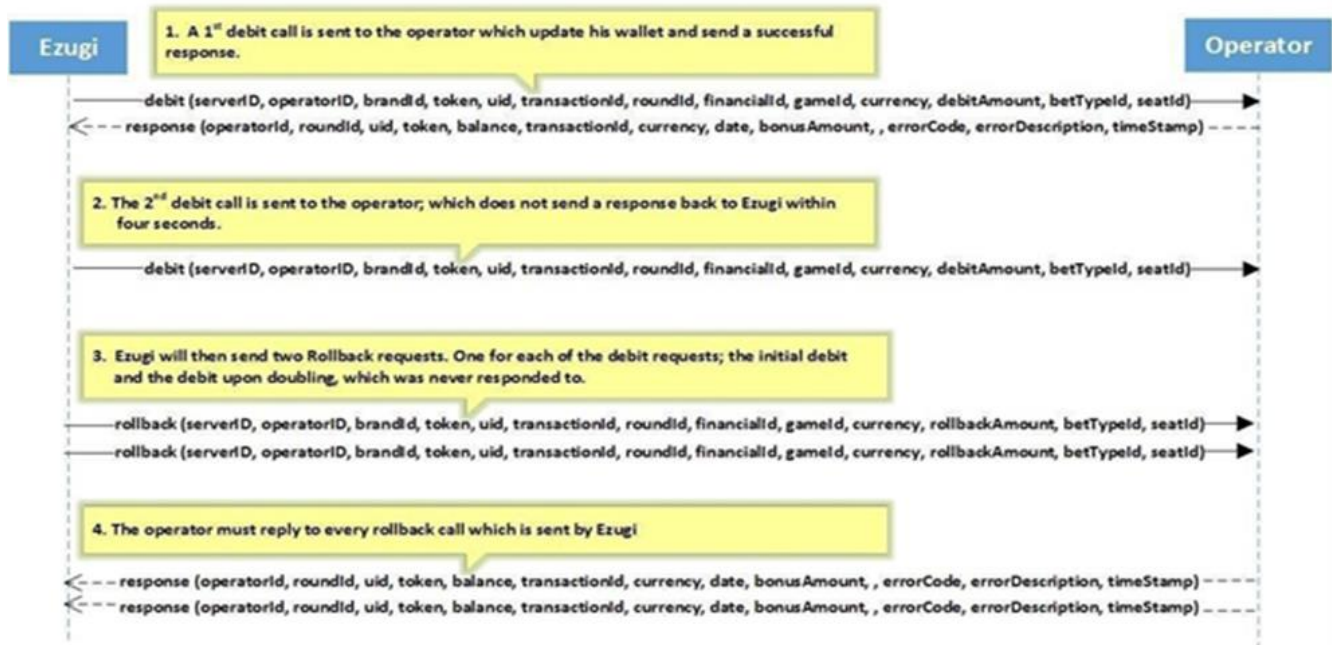
When the operator receives a rollback transaction, the main procedure should be as follows:

- If the debit transaction (with same transactionId) has been accepted by the operator (the balance was decreased), the rollback for the transaction must be processed. The balance should increase based upon the rollback amount. Please check that the rollback amount is the same as the debit amount. If not, return error code 1 and ignore the rollback call.
- If the debit transaction (with same transactionId) has NOT been accepted by the operator (the balance was not decreased), the rollback MUST be ignored to prevent the balance from unnecessarily changing. **Reply with a proper response stating that the balance has not changed. Error code 9 should be used.**
- **!!** If a rollback is received with a transactionId that does not match any stored debit transaction, the debit may still come at a later point and must be reversed. Therefore, the operator must hold onto this rollback for an hour or until a matching debit call does arrive. One way to solve this issue is by storing this transactionId along with other transactionIds that have already processed. This way, if a debit is received with the same transactionId, it will not process.

The following sequence diagram describes an example of the communication that takes place between Ezugi and the operator during a rollback event.

Rollback example -

A player makes an initial blackjack bet. When given the option, he doubles the bet causing a 2nd debit. In the event that Ezugi does not receive a response for the 2nd debit request, Ezugi will rollback the entire round for the player at the current seat (both debit calls)



5.4.2 HTTP Header

Parameter Name	Data Type/Description
hash	<p>VARCHAR (100)</p> <p>Hash Signature Check: an operator will receive on all requests coming from Ezugi a hash signature signed by a secret key that will be issued by Ezugi specifically for the operator. The hash signature will be created by using SHA256 hash algorithm and will create a signature by using a secret key and the request that is sent to the operator. The output format is base64 encoding. The operator should compare the received hash signature with his own signature that will be create by using the secret key and the received request (the signature should be removed from the received request before calculating the signature on the operator's end).</p>

5.4.3 Input Parameters

Parameter Name	Data Type/ Description
serverId	INT (11)
	A unique serverId of the Live Casino system.
operatorId	INT (11)
	Defines the particular operator that the player belongs to.
token	VARCHAR (250)
	This token is the session token; a unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session. A recommended token format for all our operators is UUID. A token should have at least 20 symbols which should be a mix of digits and characters. Any token (initial authentication or financial) issued by an operator should have an expiration date and should expire if not used after a short period of time. Financial tokens, if not used, should expire after 30-40 minutes.
uid	VARCHAR (50)
	An operator's unique identification for each player. Please only use alphanumeric characters separated by hyphen, underscore, or dot.
transactionId	Char (50)
	A unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once.
roundId	BigInt (18)
	A unique key for each serverId, to identify a game cycle. Games with more than one financial cycle will have the same roundId for all requests.
gameId	TINYINT (4)
	See section 6.1.2. Game ID codes
tableId	INT (11)
	A unique key per table ID.
currency	Char (4)
	The currency code for the balance in the current session.
rollbackAmount	Double (25,2)
	The amount requested to be added to the player's wallet. The value must be a positive number above zero and will always be equal to the debitAmount.
seatId	VARCHAR (10)
	The seat index number from where the debit occurred (Relevant only to BJ) "s1" to "s7" refers to the player's initial seat "s1-2" to "s7-2" refers to split seats "s0" refers to tips. (A tip can be placed without acquiring a seat).
platformId	Int (10)
	Is an indication for the player's platform/device. Possible values are: 0 – HTML desktop, 2 – Smartphone, 3 – Tablet
timestamp	Long
	Time representation in UNIX milliseconds format. For example: "timestamp":1531141297495

5.4.4 Output Parameters

Parameter Name	Data Type/ Description	Optional/Mandatory
operatorId	INT (11)	Mandatory
	The operator's unique identifier	
roundId	BigInt (18)	Mandatory
	A unique key per server ID, to identify a game cycle. Games with more than one financial cycle will have the same roundId for all requests.	
uid	VARCHAR (50)	Mandatory
	An operator's unique identification for each player. Please only use alphanumeric characters separated by hyphen, underscore, or dot.	
token	VARCHAR (50)	Mandatory
	<p>This token is the session token; a unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session.</p> <p>A recommended token format for all our operators is UUID. A token should have at least 20 symbols which should be a mix of digits and characters. Any token (initial authentication or financial) issued by an operator should have an expiration date and should expire if not used after a short period of time. Financial tokens, if not used, should expire after 30-40 minutes.</p>	
balance	Double (25,2)	Mandatory
	Available money in the player wallet at the time of the request. The value will be presented in the currency selected for the session	
transactionId	VARCHAR (50)	Mandatory
	The same transactionId as was sent in the request for this response. A unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once.	
currency	CHAR (4)	Mandatory
	The currency of the balance	
bonusAmount	Double (25,2)	Optional
	The bonus amount from the total rollback amount. * Cannot be more than rollback amount.	
errorCode	TINYINT (4)	Mandatory
	Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0.	
errorDescription	VARCHAR (100)	Mandatory
	A string that describes the response. This string is not the message to the player but rather details of the error.	
timestamp	VARCHAR (13)/Long	Mandatory
	<p>1. Time is represented in UNIX milliseconds format. An example and instructions: http://currentmillis.com/</p> <p>2. The Operator can return the timestamp of the balance with their own incremental counter rather than an actual timestamp.</p> <p>For example, "timestamp": "1", "timestamp": "2", "timestamp": "3", etc.</p> <p>In both cases the latest actual player's balance should come to the Ezugi system with the latest timestamp.</p>	

5.4.5 API Retransmission Policy & Error Handling

In the event of a final error response - The Live Casino server will NOT repeat the **rollback** call.

In the event of:

Socket Timeout- May occur when the connection is closed by the system.

Client's response with errorCode 10 (see p. 6.1.4. Error Codes)

By default, the Live Casino server will repeat the Rollback call three times in ten second intervals.

Please note that Ezugi can customize Rollback retries logic per operator **Custom options are:**

1. In static retries it's possible to change:

Number of retries

Interval between retries

2. In incremental retries it's possible to change to the following mechanism:

2.1 54 retries spanning over a week, increasing the time between retries each time

Period	Sending interval	Total rollback request
First minute	Every 10 seconds	6
1m-5m	Every 30 seconds	8
5m-1H	Every 5 minutes	11
1H-24H	Every 1 hour	23
1D-1W	Every day	6

2.2 14 retries for one hour, increasing the time between retries each time.

Period	Sending interval	Total rollback request
First minute	Every 10 seconds	5
1m-5m	Every 60 seconds	5
5m-8m	180 sec	1
8m-14m	360 sec	1
14m-30m	1000 sec	1
30m-1H	2000 sec	1

- All operators are expected to gracefully handle any added or unknown parameters in the request data

5.5 Credit API

5.5.1 Description

The Credit API enables the Live Casino platform to perform a financial transaction of returning money into the player's wallet through the operator's server. When calculating the results, **a player that loses has a credit amount of 0**. The player that wins is credited with the total of the payout amount.

Credit calls may occur in the middle of a game round in the following situations:

1. When a player has more than one seat playing BJ, and Ezugi receives one of the debit calls too late, then the credit will be sent with a returnReason=1.
2. In the Blackjack game, if a player decides to "Surrender", he will immediately get refunded and kicked out of the round.
3. In the Blackjack game, if a player's hand busts, a credit request with a credit amount of 0 will be sent before the following seats continue playing.

The number of credits per round (and per seat in blackjack games) must equal the number of debits sent during that round (or seat activity) individually per player. A round can be closed once the number of credits and debits are equal, **and** one of those credits has an isEndRound field with a value of true. For blackjack each seat will close with its own isEndRound=true.

5.5.2 HTTP Header

Parameter Name	Data Type/Description
hash	VARCHAR (100)
	Hash Signature Check: an operator will receive on all requests coming from Ezugi a hash signature signed by a secret key that will be issued by Ezugi specifically for the operator. The hash signature will be created by using SHA256 hash algorithm and will create a signature by using a secret key and the request that is sent to the operator. The output format is base64 encoding. The operator should compare the received hash signature with his own signature that will be create by using the secret key and the received request (the signature should be removed from the received request before calculating the signature on the operator's end).

5.5.3 Input Parameters

Parameter Name	Data Type/Description
serverId	Int (11)
	A unique serverId of the Live Casino system.
operatorId	Int (11)
	Defines the particular operator that the player belongs to.
token	VARCHAR (250)
	This token is the session token; a unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session. A recommended token format for all our operators is UUID. A token should have at least 20 symbols which should be a mix of digits and characters. Any token (initial authentication or financial) issued by an operator should have an expiration date and should expire if not used after a short period of time. Financial tokens, if not used, should expire after 30-40 minutes.
uid	VARCHAR (50)
	An operator's unique identification for every player Please only use alphanumeric characters separated by hyphen, underscore, or dot.
transactionId	Char (50)
	A unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once.
debitTransactionId	Char (50)
	Debit transaction Id of corresponding outgoing credit request. *Main
roundId	BigInt (18)
	A unique key per serverId, to identify a game cycle. Games with more than one financial cycle will have the same roundId for all requests.
gameId	TINYINT (4)
	See p.6.1.2. Game ID codes
tableId	Int (11)
	A unique key per table ID.
betTypeId	VARCHAR (3)
	The ID of the credit, in order to match the number of credits and debits.
currency	Char (4)
	The currency code of the balance.
creditAmount	Double (25.2)
	The amount requested to be credited to the player's wallet. Value >=0.

returnReason	TINYINT (4)
	Defines the reason for returning the money. Refer to section 6.1.1.
isEndRound	Boolean
	<p>Defines whether this financial transaction is the final transaction of the specific round. Will contain yes/no parameters.</p> <p>* Here Tinyint came as Boolean field type because MySql doesn't contain Boolean type</p> <p>** Please don't use this param for close round logic without a special confirmation from Ezugi integration team.</p>
gameDataString	JSON
	<p>A JSON object nested in the main JSON request with the game data history of only the specific player, public bets, moves and results.</p> <p>* This will be sent only at the end of a round.</p> <p>* Can be removed at all from the credit call</p> <p>* It's using escape characters so the parameters can be mapped as</p>
seatId	VARCHAR (10)
	<p>The seat index number from where the debit occurred (Relevant only to BJ)</p> <p>"s1" to "s7" refers to the player's initial seat</p> <p>"s1-2" to "s7-2" refers to split seats</p> <p>"s0" refers to tips. (A tip can be placed without acquiring a seat).</p>
creditIndex	VARCHAR (10)
	<p>"x y" in which Y is the total number of credits for this round (in BJ it is for a specific seat and even split seat) group, and the x is the specific credit index number (between 1.y).</p> <p>The last index (y y) is the one with isEndRound=true (if EndRound is set properly)</p>
platformId	Int (10)
	Is an indication for the player's platform/device. Possible values are: 0 – HTML desktop, 2 – Smartphone, 3 – Tablet
timestamp	Long
	Time representation in UNIX milliseconds format. For example: "timestamp":1531141297495

5.5.4 Output Parameters

Parameter Name	Data Type/Description	Optional/Mandatory
operatorId	INT (11)	Mandatory
	An operator's unique identifier	
roundId	BigInt (18)	Mandatory
	A unique key per server ID, to identify a game cycle. Games with more than one financial cycle will have the same roundId for all requests.	
uid	VARCHAR (50)	Mandatory
	An operator's unique identification for every player Please only use alphanumeric characters separated by hyphen, underscore, or dot.	
token	VARCHAR (250)	Mandatory
	This token is the session token; a unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session. A recommended token format for all our operators is UUID. A token should have at least 20 symbols which should be a mix of digits and characters. Any token (initial authentication or financial) issued by an operator should have an expiration date and should expire if not used after a short period of time. Financial tokens, if not used, should expire after 30-40 minutes.	
balance	Double (25,2)	Mandatory
	Available money in the player's wallet at the time of the request. The value will be presented in the currency of the session.	
transactionId	VARCHAR (50)	Mandatory
	The same transactionId as was sent in the request for this response. A unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once.	
currency	CHAR (4)	Mandatory
	The currency of the balance	
bonusAmount	Double (25,2)	Optional
	The bonus amount from the total credit amount. * Cannot be more than credit amount.	
errorCode	TINYINT (4)	Optional
	Whether or not the request has been processed successfully. If no error is detected, the code value is 0.	
errorDescription	VARCHAR (100)	Mandatory
	A string that describes the response. This string is not the message to the player but rather details of the error.	
timestamp	VARCHAR (13)	Mandatory
	<ol style="list-style-type: none"> Time is represented in UNIX milliseconds format. An example and instructions: http://currentmillis.com/ The Operator can return the timestamp of the balance with their own incremental counter rather than an actual timestamp. For example, "timestamp": "1", "timestamp": "2", "timestamp": "3", etc. In both cases the latest actual player's balance should come to the Ezugi system with the latest timestamp. 	

5.5.5 API Retransmission Policy & Error Handling

In the event of a final error response- The Live Casino server will NOT repeat the call. A relevant message is presented to the user.

In the event of:

- a. **Socket Timeout-** May occur when the connection is closed by the system.
- b. **Client's response with errorCode 10** (see p. 6.1.4. Error Codes)

The Live Casino server will repeat the Credit call three times in ten second intervals.

Please note that Ezugi can customize credit retries logic per operator

Custom options are:

1. In static retries it's possible to change:

- 1.1 Number of retries
- 1.2 Interval between retries

2. In incremental retries it's possible to change to the following mechanism:

2.1 54 retries spanning over a week, increasing the time between retries each time

Period	Sending interval	Total Credit Request
First minute	Every 10 seconds	6
1m-5m	Every 30 seconds	8
5m-1H	Every 5 minutes	11
1H-24H	Every 1 hour	23
1D-1W	Every day	6

2.2 14 retries for one hour, increasing the time between retries each time.

Period	Sending interval	Total Credit Request
First minute	Every 10 seconds	5
1m-5m	Every 60 seconds	5
5m-8m	180 sec	1
8m-14m	360 sec	1
14m-30m	1000 sec	1
30m-1H	2000 sec	1

All operators are expected to gracefully handle any added or unknown parameters in the request data.

5.6 Get New Token

5.6.1 Description

This API is relevant for operators that cannot have the same token for different games and for when they are going to open Ezugi games from within another game or lobby. For example, when using mobile lobby in which the post message option is not working, in this case Ezugi needs to get a new valid token from the operator when launching a game.

5.6.2 Input Parameters

Parameter Name	Data Type/Description
hash	VARCHAR (100)
	Hash Signature Check: an operator will receive on all requests coming from Ezugi a hash signature signed by a secret key that will be issued by Ezugi specifically for the operator. The hash signature will be created by using SHA256 hash algorithm and will create a signature by using a secret key and the request that is sent to the operator. The output format is base64 encoding. The operator should compare the received hash signature with his own signature that will be create by using the secret key and the received request (the signature should be removed from the received request before calculating the signature on the operator's end).

5.6.3 Input Parameters

Parameter Name	Data Type/ Description
operatorId	Int (11)
	Defines the operator that the player belongs to. This is allocated by the Live Casino system in advance.
currentToken	VARCHAR (250)
	Current game token (not web token) that is used by the operator for this player.
gameId	TINYINT (4)
	See p.6.1.2. Game ID codes.
tableId	Int
	A unique key per table ID.
uid	VARCHAR (50)
	An operator's unique identification for every player
timestamp	Long
	Time representation in UNIX milliseconds format. For example: "timestamp":1531141297495

5.6.4 Output Parameters

Parameter Name	Data Type/ Description	Optional/Mandatory
operatorId	INT (11)	Mandatory
	Defines the operator that the player belongs to. This is allocated by the Live Casino system in advance.	
token	VARCHAR (250)	Mandatory
	This token is the session token; a unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session.	
balance	Double (25,2)	Mandatory
	Available money in the player's wallet at the time of the request. The value will be presented in the currency of that session.	
uid	VARCHAR (50)	Mandatory
	An operator's unique identification for every player	
timestamp	Varchar (13) /long	Mandatory
	1. Time is represented in UNIX milliseconds format. An example and instructions: http://currentmillis.com/ 2. The Operator can return the timestamp of the balance with their own incremental counter rather than an actual timestamp. For example, "timestamp": "1", "timestamp": "2", "timestamp": "3", etc. In both cases the latest actual player's balance should come to the Ezugi system with the latest timestamp.	
errorCode	TINYINT (4)	Mandatory
	Identifies whether the request has been processed successfully. If no error is detected, this code value is 0.	
errorDescription	VARCHAR (100)	Mandatory
	A string that describes the response. This string is not the message to the player but rather details of the error.	

5.6.5 API Retransmission Policy & Error Handling

- In the event of a final error response - The Live Casino server will NOT repeat the Get New Token call.
- Possible exceptions:
 - a. **Socket Timeout**- May occur when the connection is closed by the system.
 - b. **Client's response with errorCode 10** (see p. 6.1.4. Error Codes)

The Live Casino server will repeat the Get New Token call four more times. If still unsuccessful, the game will not load.

All operators are expected to gracefully handle any added or unknown parameters in the request data.

6. Dictionaries and Naming Convention

6.1 Dictionaries

6.1.1 Return reason codes

returnReason	Description
0	Successful bet
1	Cancel bet
2	Canceled round

6.1.2 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
13	Lucky 7
14	Sic BO
15	Casino Hold'em
16	Bet on Teen Patti
17	Three Card Poker (NJ) / Teen Patti
18	Roulette with JP
19	32 Cards
20	Baccarat KO
21	Baccarat Super 6
24	Dragon Tiger
25	No Commission Baccarat
26	Baccarat Dragon Bonus
27	Baccarat Queenco
28	Baccarat Punto Banco
29	Roulette Portomaso

30	Bet on Roulette
31	American Roulette
32	Triple Roulette
38	Andar Bahar
39	OTT Andar Bahar
43	One Day Teen Patti
44	Auto Sic Bo
45	Cricket War
46	BJ Salon Prive
47	Dream catcher (retail)
48	EZ Dealer Roulette
49	Live slots
50	One Day Teen Patti Back & Lay
51	Video Blackjack
52	Ultimate Sic Bo
53	Royal Poker
54	Ultimate Roulette
55	Ultimate Andar Bahar

Note:

*New games and features are constantly added. To support these changes, please ensure the gameld field can accept any value between 1 and 99. When a new game is added, a description of the gameld will be provided as well. **Not all the games on the list are available in production.**

6.1.3 Bet Type codes

Debit betTypeID	Credit betTypeID	Can have isEndRound=true	Can have amount >0 credit	Description	Games
GENERAL BET					
1	101	Yes	Yes	Main Bet & Credit	All games
3	103 (optional)	N/A	No (optional)	TIP	
BLACKJACK GAMES					
4	104	No	If dealer has BJ	INSURANCE	Blackjack American Blackjack Unlimited BJ
5	105	No	Only with returnReason 1 or 2	DOUBLE	Blackjack American Blackjack Unlimited BJ
6	106	Yes	Yes	SPLIT**	Blackjack American Blackjack Unlimited BJ
POKER GAMES					
1	101	Yes	Yes	Main Bet & Credit (Sent for ANTE call)	All games
24	124	Yes	Yes	CALL	Casino Hold'em Teen Patti Three Card Poker (NJ)
ROYAL POKER GAMES					
1	101	Yes	Yes	Main Bet & Credit (Sent for ANTE call)	All games
8	108	Yes	No	Swap	
9	109	Yes	No	Buy Player Card	
10	110	Yes	Yes	Royal Poker Insurance	
11	111	Yes	No	Buy Dealer Card	
24	124	Yes	Yes	Call	
JACKPOT (Temporarily terminated)					
27	127	No	Yes	Jackpot	Roulette only

* TIP bet is a special type of debit that is not a part of the game's financial actions (bets and "win/lose"). This parameter is used for correct tip handling. It does not start or end any round and could take place in between rounds as well. When playing blackjack, the player may only tip once he has a seat in order for it to be clear who the tip belongs to.

** The SPLIT Debit betTypeID is associated with S_X , and the Credit betTypeID is associated with S_{X-2} , where X is the seat name (seat ID) from which the seat was split.

6.1.4 Error Codes

errorCode	Error Description
0	Completed successfully
1	General error
2	Saved for future use (should not be used)
3	Insufficient funds
4	Operator limit to the player 1 (can be set as Bet Limit/Session Limit/Loss limit etc.)
5	Operator limit to the player 2 (can be set as Bet Limit/Session Limit/Loss limit etc.)
6	Token not found
7	User not found
8	User blocked
9	Transaction not found
10	Transaction timed out <i>*Ezugi system can repeat the call if error code 10 is sent by operator with authentication, credit, and rollback responses according to operator's retries settings. The debit transaction will be converted into a rollback transaction.</i>
11	Real balance is not enough for tipping

Note: If the same **transactionId** was sent more than once, you must send us the same response with the original error code. In the error description write – “transaction has already proceeded” – Balance should not be modified.

6.1.5 Languages Dictionary (iso 639-1)

language	Language name
en	English
es	Spanish
ru	Russian
zf	Simplified Chinese
zh	Traditional Chinese
hi	Hindi
vi	Vietnamese
ja	Japanese
ko	Korean

Note: Any two-letter language code can be used. If the code does not exist in our system, the default language, English, will load.

6.1.6 Currencies Dictionary (iso 4217)

currencyCode	Currency name
GBP	United Kingdom Pound
USD	United States Dollar
EUR	Euro
Any three-letter currency code can be used if it is registered in our system.	

6.2 Naming Conventions

6.2.1 Field Key Names for All API Calls

- All keys will be in lower case, except the complex of two or more words.
- From the second word and on (3rd, 4th, etc.), words will start with an upper case (capital) letter.
- The rest of the letters should be lower case.
- For example:
 - token
 - balance
 - transactionId
 - errorCode
 - errorDescription
 - betTypeId

7. Integration into Live Casino Staging Environment

7.1 Integration Server Details:

Launch URL: <https://playint.tableslive.com/auth/>

IPs: 109.97.118.250, 109.102.212.194

7.2 Launching Lobby and Games:

The following is a link to open the HTML lobby or a game directly. The bold section of the link is necessary to open the lobby while the following section, enclosed in square brackets, may be added to open a game directly.

[https://playint.tableslive.com/auth/?token={generatedValidPlayerToken}&operatorId={generatedByLiveCasinoSide}&\[language={2letterISOcode}&clientType=html5&openTable={gameTableId}&tableLimits={tableLimitId}&homeUrl=\[http/https\]\[DOMAIN\]?\[KEY=VALUE\]%26\[KEY=VALUE\]%26\[KEY=VALUE\]%26.....\]](https://playint.tableslive.com/auth/?token={generatedValidPlayerToken}&operatorId={generatedByLiveCasinoSide}&[language={2letterISOcode}&clientType=html5&openTable={gameTableId}&tableLimits={tableLimitId}&homeUrl=[http/https][DOMAIN]?[KEY=VALUE]%26[KEY=VALUE]%26[KEY=VALUE]%26.....])

7.3 Game Size and Ratio:

Aspect ratio for a new browsing window/iframe is 16:9. Minimum height: 520

8. Appendix

8.1 Production Studios Information

Production Studio documents will be sent separately. Please contact the integration team.

8.2 Examples

8.2.1 Authentication Example

Request:

```
{
  "platformId": 0,
  "operatorId": 10000000,
  "token": "*****dbcd",
  "timestamp": 1668765433118
}
```

Response:

```
{
  "operatorId": 10000000,
  "uid": "57269",
  "nickName": "Christopher Columbus",
  "token": "b7466383b26c24946996a1c5a5f16f2d",
  "playerTokenAtLaunch": "*****dbcd",
  "balance": 422.05,
  "currency": "USD",
  "language": "en",
  "date": "2022-11-18 09:57:13.186482",
  "clientIP": "127.0.0.1",
  "VIP": "0",
  "errorCode": 0,
  "errorDescription": "ok",
  "timestamp": "1668765433187"
}
```

8.2.2 Get New Token Example

Request:

```
{
  "currentToken": "a27b1951-8193-44ac-ba4c-795df3325a97",
  "gameId": 3,
  "uid": "57269",
  "tableId": 601000,
  "operatorId": 10000000,
  "timestamp": 1668668452420
}
```

Response:

```
{
  "errorCode": 0,
  "errorDescription": "ok",
  "operatorId": 10000000,
  "token": "d484a09d-be6e-4330-8b8c-81bb265d1e11",
  "balance": 221.18,
  "uid": "1599-MABU886165371I",
  "timestamp": 1668668453063
}
```

8.3 Examples Per Game

8.3.1 Blackjack

Debit

Request:

```
{
  "gameId": 1,
  "debitAmount": 5.00,
  "platformId": 0,
  "serverId": 102,
  "transactionId": "d6ad62ae-de5c-4de1-8516-9d3da3af3820",
  "token": "*****6f2d",
  "uid": "57269",
  "betTypeID": 1,
  "tableId": 1,
  "seatId": "s6",
  "currency": "USD",
  "operatorId": 10000000,
  "roundId": 54680447,
  "timestamp": 1668766288973
}
```

Response:

```
{
  "uid": "57269",
  "operatorId": 10000000,
  "nickName": "Christopher Columbus",
  "token": "*****6f2d",
  "balance": 422.05,
  "transactionId": "d6ad62ae-de5c-4de1-8516-9d3da3af3820",
  "timestamp": "1668766289055",
  "currency": "USD",
  "errorCode": 0,
  "roundId": 54680447,
  "bonusAmount": 1,
  "errorDescription": "OK"
}
```

*Credit:**Request:*

```
{
  "gameId": 1,
  "debitTransactionId": "d6ad62ae-de5c-4de1-8516-9d3da3af3820",
  "isEndRound": true,
  "creditIndex": "1|1",
  "platformId": 0,
  "serverId": 102,
  "transactionId": "c6ad62ae-de5c-4de1-8516-9d3da3af3820",
  "token": "*****6f2d",
  "uid": "57269",
  "returnReason": 0,
  "betTypeId": 101,
  "tableId": 1,
  "seatId": "s6",
  "currency": "USD",
  "creditAmount": 0.00,
  "operatorId": 10000000,
  "roundId": 54680447,
  "timestamp": 1668766330157
}
```

Response:

```
{
  "operatorId": 10000000,
  "uid": "57269",
  "nickName": "Christopher Columbus",
  "token": "*****6f2d",
  "balance": 422.05,
  "transactionId": "c6ad62ae-de5c-4de1-8516-9d3da3af3820",
  "currency": "USD",
  "timestamp": "1668766330227",
  "errorCode": 0,
  "roundId": 54680447,
  "bonusAmount": 0,
  "errorDescription": "ok"
}
```

8.3.2 Baccarat

Debit

Request:

```
{
  "gameId": 2,
  "debitAmount": 8,
  "platformId": 0,
  "serverId": 102,
  "transactionId": "de4c5e74-0efc-45f4-9161-eddfda85ff18",
  "token": "*****25f6",
  "uid": "57369",
  "betTypeID": 1,
  "tableId": 100,
  "currency": "USD",
  "operatorId": 10000000,
  "roundId": 54681908,
  "timestamp": 1668767899835
}
```

Response:

```
{
  "uid": "57369",
  "operatorId": 10000000,
  "nickName": "Queen Victoria",
  "token": "*****25f6",
  "balance": 13842774.19,
  "transactionId": "de4c5e74-0efc-45f4-9161-eddfda85ff18",
  "timestamp": "1668767899905",
  "currency": "USD",
  "errorCode": 0,
  "roundId": 54681908,
  "bonusAmount": 1,
  "errorDescription": "OK"
}
```

Credit

Request:

```
{
  "gameId": 2,
  "debitTransactionId": "de4c5e74-0efc-45f4-9161-eddfda85ff18",
  "isEndRound": true,
  "creditIndex": "1|1",
  "platformId": 0,
  "serverId": 102,
  "transactionId": "ce4c5e74-0efc-45f4-9161-eddfda85ff18",
  "token": "*****25f6",
  "uid": "57369",
  "returnReason": 0,
  "betTypeId": 101,
  "tableId": 100,
  "currency": "USD",
  "creditAmount": 3.54,
  "operatorId": 10000000,
  "roundId": 54681908,
  "timestamp": 1668767917161
}
```

Response:

```
{
  "operatorId": 10000000,
  "uid": "57369",
  "nickName": "Queen Victoria",
  "token": "*****25f6",
  "balance": 13842777.73,
  "transactionId": "ce4c5e74-0efc-45f4-9161-eddfda85ff18",
  "currency": "USD",
  "timestamp": "1668767917243",
  "errorCode": 0,
  "roundId": 54681908,
  "bonusAmount": 0,
  "errorDescription": "ok"
}
```

8.3.3 Casino Hold'em

Debit (Ante)

Request:

```
{
  "gameId": 15,
  "debitAmount": 4,
  "platformId": 0,
  "serverId": 102,
  "transactionId": "d5694d06-5492-491f-b45b-38eedfbd39a3",
  "token": "*****25f6",
  "uid": "57369",
  "betTypeID": 1,
  "tableId": 507000,
  "currency": "USD",
  "operatorId": 10000000,
  "roundId": 54682069,
  "timestamp": 1668768079119
}
```

Response:

```
{
  "uid": "57369",
  "operatorId": 10000000,
  "nickName": "Queen Victoria",
  "token": "*****25f6",
  "balance": 13842771.73,
  "transactionId": "d5694d06-5492-491f-b45b-38eedfbd39a3",
  "timestamp": "1668768079203",
  "currency": "USD",
  "errorCode": 0,
  "roundId": 54682069,
  "bonusAmount": 0,
  "errorDescription": "OK"
}
```

*Debit (Call):**Request*

```
{
  "gameId": 15,
  "debitAmount": 6,
  "platformId": 0,
  "serverId": 102,
  "transactionId": "d2f4ef22-1700-4b68-a395-a5f5c5b74035",
  "token": "*****25f6",
  "uid": "57369",
  "betTypeId": 24,
  "tableId": 507000,
  "currency": "USD",
  "operatorId": 10000000,
  "roundId": 54682069,
  "timestamp": 1668768095065
}
```

Response:

```
{
  "uid": "57369",
  "operatorId": 10000000,
  "nickName": "Queen Victoria",
  "token": "*****25f6",
  "balance": 13842765.73,
  "transactionId": "d2f4ef22-1700-4b68-a395-a5f5c5b74035",
  "timestamp": "1668768095173",
  "currency": "USD",
  "errorCode": 0,
  "roundId": 54682069,
  "bonusAmount": 1,
  "errorDescription": "OK"
}
```

*Credit (Ante):**Request*

```
{
  "gameId": 15,
  "debitTransactionId": "d2f4ef22-1700-4b68-a395-a5f5c5b74035",
  "gameDataString": "",
  "isEndRound": false,
  "creditIndex": "1|2",
  "platformId": 0,
  "serverId": 102,
  "transactionId": "c2f4ef22-1700-4b68-a395-a5f5c5b74035",
  "token": "*****25f6",
  "uid": "57369",
  "returnReason": 0,
  "betTypeID": 124,
  "tableId": 507000,
  "currency": "USD",
  "creditAmount": 0,
  "operatorId": 10000000,
  "roundId": 54682069,
  "timestamp": 1668768109189
}
```

Response:

```
{
  "operatorId": 10000000,
  "uid": "57369",
  "nickName": "Queen Victoria",
  "token": "*****25f6",
  "balance": 13842765.73,
  "transactionId": "c2f4ef22-1700-4b68-a395-a5f5c5b74035",
  "currency": "USD",
  "timestamp": "1668768109301",
  "errorCode": 0,
  "roundId": 54682069,
  "bonusAmount": 0,
  "errorDescription": "ok"
}
```


Credit (Call):

Request

```
{
  "gameId": 15,
  "debitTransactionId": "d5694d06-5492-491f-b45b-38eedfbd39a3",
  "isEndRound": true,
  "creditIndex": "2|2",
  "platformId": 0,
  "serverId": 102,
  "transactionId": "c5694d06-5492-491f-b45b-38eedfbd39a3",
  "token": "*****25f6",
  "uid": "57369",
  "returnReason": 0,
  "betTypeId": 101,
  "tableId": 507000,
  "currency": "USD",
  "creditAmount": 0,
  "operatorId": 10000000,
  "roundId": 54682069,
  "timestamp": 1668768109332
}
```

Response:

```
{
  "operatorId": 10000000,
  "uid": "57369",
  "nickName": "Queen Victoria",
  "token": "*****25f6",
  "balance": 13842765.73,
  "transactionId": "c5694d06-5492-491f-b45b-38eedfbd39a3",
  "currency": "USD",
  "timestamp": "1668768109426",
  "errorCode": 0,
  "roundId": 54682069,
  "bonusAmount": 0,
  "errorDescription": "ok"
}
```

8.3.4 Tip example

Request:

```
{
  "gameId": 1,
  "debitAmount": 5.00,
  "platformId": 0,
  "serverId": 102,
  "transactionId": "d3cf7d56-d89e-4b6d-a64e-f222e7784e32",
  "token": "*****6f2d",
  "uid": "57269",
  "betTypeId": 3,
  "tableId": 1,
  "seatId": "s0",
  "currency": "USD",
  "operatorId": 10000000,
  "roundId": 54681005,
  "timestamp": 1668766897868
}
```

Response:

```
{
  "uid": "57269",
  "operatorId": 10000000,
  "nickName": "Christopher Columbus",
  "token": "*****6f2d",
  "balance": 4222254884.05,
  "transactionId": "d3cf7d56-d89e-4b6d-a64e-f222e7784e32",
  "timestamp": "1668766897952",
  "currency": "USD",
  "errorCode": 0,
  "roundId": 54681005,
  "bonusAmount": 1,
  "errorDescription": "OK"
}
```

8.3.5 GameDataString Example

Credit Request:

```
{
  "gameId": 1,
  "debitTransactionId": "d6f7f9db-ce29-4a50-95fa-573e1a6235a1",
  "gameDataString": "{\\\"TableId\\\":\\\"1\\\",\\\"PlayerId\\\":\\\"10000000\\\"57269\\\"1\\\",\\\"CardInShoe\\\":98,\\\"DealerCardHandValue\\\":\\\"20\\\",\\\"BetType\\\":101,\\\"BetAmount\\\":10.0,\\\"InsuranceDecision\\\":\\\"Unknown\\\",\\\"SessionCurrency\\\":\\\"USD\\\",\\\"PlayerCards\\\":[{\\\"CardName\\\":\\\"6d\\\",\\\"CardValue\\\":6},{\\\"CardName\\\":\\\"8h\\\",\\\"CardValue\\\":8},{\\\"CardName\\\":\\\"2d\\\",\\\"CardValue\\\":2}],\\\"TakenSeatsNumber\\\":1,\\\"OperatorId\\\":\\\"10000000\\\",\\\"DealerCards\\\":[{\\\"CardName\\\":\\\"Tc\\\",\\\"CardValue\\\":10},{\\\"CardName\\\":\\\"Jc\\\",\\\"CardValue\\\":10}],\\\"PlayerCardHandValue\\\":\\\"16\\\",\\\"BetsList\\\":[{\\\"BetName\\\":\\\"RegularBet\\\",\\\"BetAmount\\\":5.0},{\\\"BetName\\\":\\\"DoubleBet\\\",\\\"BetAmount\\\":5.0}],\\\"ServerId\\\":102,\\\"Version\\\":\\\"1.3\\\",\\\"DealerId\\\":\\\"20\\\",\\\"GameResults\\\":\\\"Lose\\\",\\\"roundId\\\":54680952,\\\"WinAmount\\\":0.0,\\\"WinningBets\\\":{\\\"GameId\\\":1,\\\"SeatId\\\":\\\"s6\\\"}}",
  "isEndRound": true,
  "creditIndex": "2|2",
  "platformId": 0,
  "serverId": 102,
  "transactionId": "c6f7f9db-ce29-4a50-95fa-573e1a6235a1",
  "token": "*****6f2d",
  "uid": "57269",
  "returnReason": 0,
  "betTypeID": 101,
  "tableId": 1,
  "seatId": "s6",
  "currency": "USD",
  "creditAmount": 0.00,
  "operatorId": 10000000,
  "roundId": 54680952,
  "timestamp": 1668766873061
}
```

Credit Response:

```
{
  "operatorId": 10000000,
  "uid": "57269",
  "nickName": "Christopher Columbus",
  "token": "*****6f2d",
  "balance": 4222254894.05,
  "transactionId": "c6f7f9db-ce29-4a50-95fa-573e1a6235a1",
  "currency": "USD",
  "timestamp": "1668766873141",
  "errorCode": 0,
  "roundId": 54680952,
  "bonusAmount": 0,
  "errorDescription": "ok"
}
```

8.3.6 Rollback Example

Rollback Request:

```
{
  "operatorId": 10000000,
  "uid": "57369",
  "transactionId": "b36568a5-b34b-4b34-9ea6-1df7e89c63c9",
  "gameId": 1,
  "token": "*****25f5",
  "rollbackAmount": 1.99,
  "betTypeID": 1,
  "serverId": 102,
  "roundId": 836944,
  "currency": "USD",
  "seatId": "s3",
  "platformId": 0,
  "tableId": 1,
  "timestamp": 1666767917161
}
```

Rollback Response:

```
{
  "errorCode": 0,
  "errorDescription": "Completed successfully",
  "timestamp": 1666767917163,
  "operatorId": 10000000,
  "roundId": 836944,
  "uid": "57369",
  "token": "*****25f5",
  "balance": 278.88,
  "transactionId": "b36568a5-b34b-4b34-9ea6-1df7e89c63c9",
  "currency": "USD"
}
```

8.3.7 Hash Signature Examples

Authentication

Request

Hash: 66zs3Gs+EGnPJdtJVhVOodVCBvC0qte/BakED74BYM=

```
{
  "operatorId": 10000000,
  "uid": "57369",
  "transactionId": "154e14d5-e593-48f3-8a38-220311b8c9e7",
  "gameId": 1,
  "token": "*****25f9",
  "debitAmount": 1.99,
  "betTypeId": 1,
  "serverId": 102,
  "roundId": 836961,
  "currency": "USD",
  "seatId": "s3",
  "platformId": 0,
  "tableId": 1,
  "timestamp": 1668768109332
}
```

Operator's response:

```
{
  "errorCode": 0,
  "errorDescription": "OK",
  "timestamp": 1668768109334,
  "operatorId": 10000000,
  "uid": "57369",
  "nickName": "Queen Victoria",
  "token": "*****25f9",
  "balance": 0,
  "currency": "USD",
  "roundId": 836961,
  "transactionId": "154e14d5-e593-48f3-8a38-220311b8c9e7"
}
```

8.3.8 Error Code Examples

Error Code 1:

Debit Request:

```
{
  "operatorId": 10000000,
  "uid": "1277335",
  "transactionId": "d6b81386-7255-4765-b3a3-e06e7f701a3d",
  "gameId": 1,
  "token": "ca1aa6b39a4c493aea55218008923d9d",
  "debitAmount": -0.11,
  "betTypeId": 1,
  "serverId": 102,
  "roundId": 712982,
  "currency": "EUR",
  "seatId": "s4",
  "platformId": 0,
  "tableId": 1,
  "timestamp": 1652444755924
}
```

Debit Response:

```
{
  "operatorId": 10000000,
  "uid": "1277335",
  "token": "ca1aa6b39a4c493aea55218008923d9d",
  "balance": 8.65,
  "transactionId": "d6b81386-7255-4765-b3a3-e06e7f701a3d",
  "currency": "EUR",
  "errorCode": 1,
  "roundId": "712982",
  "bonusAmount": 0,
  "errorDescription": "Negative amount",
  "timestamp": 1652444755967
}
```

Error Code 3:

Debit Request:

```
{
  "operatorId": 10000000,
  "uid": "812425",
  "transactionId": "dc1dd1bb-a0f0-4bd6-b65b-0ad71f42eb8e",
  "gameId": 1,
  "token": "af399bdd-2751-48cf-b7de-027bfcc014f5",
  "debitAmount": 11475.69,
  "betTypeId": 1,
  "serverId": 102,
  "roundId": 45549,
  "currency": "ZAR",
  "seatId": "s1",
  "platformId": 0,
  "tableId": 1,
  "timestamp": 1639741178710
}
```

Debit Response:

```
{
  "operatorId": 10000000,
  "roundId": 45549,
  "uid": "812425",
  "nickName": "Ezugi",
  "token": "af399bdd-2751-48cf-b7de-027bfcc014f5",
  "balance": 11475.68,
  "transactionId": "dc1dd1bb-a0f0-4bd6-b65b-0ad71f42eb8e",
  "currency": "ZAR",
  "errorCode": 3,
  "errorDescription": "Insufficient funds",
  "timestamp": 1639741179047
}
```

*Error Code 6:**Debit Request:*

```
{
  "operatorId": 10000000,
  "uid": "812425",
  "transactionId": "d04f753d-901a-4e21-8d4e-78b605ab9796",
  "gameId": 1,
  "token": "af399bdd-2751-48cf-b7de-027bfcc014f511",
  "debitAmount": 1.99,
  "betTypeId": 1,
  "serverId": 102,
  "roundId": 45550,
  "currency": "ZAR",
  "seatId": "s1",
  "platformId": 0,
  "tableId": 1,
  "timestamp": 1639741179111
}
```

Debit Response:

```
{
  "operatorId": 10000000,
  "roundId": 45550,
  "uid": "812425",
  "token": "af399bdd-2751-48cf-b7de-027bfcc014f511",
  "balance": 0.0,
  "transactionId": "d04f753d-901a-4e21-8d4e-78b605ab9796",
  "currency": "ZAR",
  "errorCode": 6,
  "errorDescription": "Token not found",
  "timestamp": 1639741179344
}
```


*Error Code 7:**Debit Request:*

```
{
  "operatorId": 10000000,
  "uid": "81242511",
  "transactionId": "db5d674b-2811-4841-9039-3fc26e44b7ec",
  "gameId": 1,
  "token": "af399bdd-2751-48cf-b7de-027bfcc014f5",
  "debitAmount": 1.99,
  "betTypeId": 1,
  "serverId": 102,
  "roundId": 45551,
  "currency": "ZAR",
  "seatId": "s1",
  "platformId": 0,
  "tableId": 1,
  "timestamp": 1639741179385
}
```

Debit Response:

```
{
  "operatorId": 10000000,
  "roundId": 45551,
  "uid": "81242511",
  "token": "af399bdd-2751-48cf-b7de-027bfcc014f5",
  "balance": 0.0,
  "transactionId": "db5d674b-2811-4841-9039-3fc26e44b7ec",
  "currency": "ZAR",
  "errorCode": 7,
  "errorDescription": "User not found",
  "timestamp": 1639741179613
}
```

*Error Code 9:**Rollback Request:*

```
{
  "operatorId": 10000000,
  "uid": "812425",
  "transactionId": "d4234ec9-e537-4414-9cd0-cbc87239207c",
  "gameId": 1,
  "token": "af399bdd-2751-48cf-b7de-027bfcc014f5",
  "rollbackAmount": 1.99,
  "betTypeId": 1,
  "serverId": 102,
  "roundId": 45546,
  "currency": "ZAR",
  "seatId": "s1",
  "platformId": 0,
  "tableId": 1,
  "timestamp": 1639741175995
}
```

Rollback Response:

```
{
  "operatorId": 10000000,
  "roundId": 45546,
  "uid": "812425",
  "token": "af399bdd-2751-48cf-b7de-027bfcc014f5",
  "balance": 11476.68,
  "transactionId": "d4234ec9-e537-4414-9cd0-cbc87239207c",
  "currency": "ZAR",
  "errorCode": 9,
  "errorDescription": "Transaction not found",
  "timestamp": 1639741176233
}
```