

Evolution Live Casino Integration Manual (One Wallet / Seamless)

Exported on 17/8/2018

User Authentication

1 Overview

User Authentication API provides an interface which can be used to create new player and/or login existing player, create game session and launch the Evolution Live Casino.

1.1 Invocation details

- Supported methods: POST
- Response format: JSON
- User Authentication 2.0 API service endpoint URL should never be revealed to the player and placed on Licensee website. User authentication request must be posted in server-to-server mode.
- User Authentication 2.0 API service endpoint URL must always use 'domain name' instead of IP address
- User Authentication 2.0 API is served over HTTPS - to ensure security and data privacy, unencrypted HTTP is not supported
- All IP addresses that need access to User Authentication 2.0 API must be white-listed in advance

1.2 Response codes and errors

User Authentication 2.0 API uses standard RFC 2616 HTTP response codes to indicate the success or failure of an API request:

- Codes in the 2xx range indicate success
- Codes in the 4xx range indicate a problem with the request
- Codes in the 5xx range indicate an error with Evolution's servers

2 User Authentication API service endpoint

`https://<hostname>/ua/v1/{casino.key}/{api.token}`

In order to launch Evolution Live Casino, Licensee has to send user authentication request to the service endpoint URL (see above). User authentication request body must be sent in JSON format using POST method. The default behavior for a successful request is to redirect player to the Evolution Live Casino page based on the parameters provided in the request body. An authentication token string will be appended to the returned URL to identify the player's session in Evolution system. After receiving player's request for returned URL, Evolution system will add the authentication token to a cookie, therefore player needs to ensure to have cookies enabled

on his side.

URL parameters:

Name	Description	Mandatory/ Optional
casino.key	Casino key for User Authentication service, provided by Evolution	M
api.token	API token for User Authentication service, provided by Evolution	M
hostname	Licensee hostname, provided by Licensee. Licensee must provide SSL(s) for hostname(s) of production game launch host(s). Evolution installs SSL and provides CNAME to configure for hostname	M

User authentication request body must be sent in following format:

```
{
  "uuid": "unique request identifier",
  "player": {
    "id": "a1a2a3a4",
    "update": true,
    "firstName": "firstName",
    "lastName": "lastName",
    "nickname": "nickname",
    "country": "DE",
    "language": "de",
    "currency": "EUR",
    "session": {
      "id": "111ssss3333rrrrr45555",
      "ip": "192.168.0.1"
    }
  },
  "config": {
    "brand": {
      "id": "1",
      "skin": "1"
    },
    "game": {
      "category": "roulette",
      "interface": "view1",
      "table": {
        "id": "vip-roulette-123"
      }
    }
  },
}
```

```

"channel": {
  "wrapped": false,
  "mobile": false
},
"urls": {
  "cashier": "http://www.chs.ee",
  "responsibleGaming": "http://www.RGam.ee",
  "lobby": "http://www.lobb.ee",
  "sessionTimeout": "http://www.sesstm.ee"
}
}
}

```

3 User Authentication 2.0 API request parameters

✂ All parameters are case sensitive.

Name				Data type	Description	Mandatory / Optional
uuid				String	Unique request id, that identifies concrete user authentication request (attempt)	M
player				Object	Object containing player details	M
	id			String (50)	Player's ID. Unique identifier of a player, assigned by Licensee	M
	update			Boolean	Indicates if player details should be updated. `True` if system is asked to update player records. `False` if player data is relevant for current session only. Updates `firstName`, `lastName`, `nickname`, `country`, `language` values.	M
	firstName			String (50)	Player's first name	M
	lastName			String (50)	Player's last name	M
	nickname			String (35)	Player's preferred nickname (screen name). Must be unique per	O

					casino instance. If not passed then player will be prompted to enter nickname on first Live casino visit	
	country			String (2)	Player's country code (ISO 3166, 2 letter code)	M
	language			String (2)	Player's preferred language (ISO 639-1, 2 letter code)	M
	currency			String (3)	Player's currency (ISO 4217, 3 letter code)	M
	session			Object	Object containing player session details	M
		id		String	Player's session ID, assigned by Licensee	M
		ip		String (15)	Player's session IP address	M
config				Object	Object containing game launch configuration options	M
	brand			Object	Object containing branding configuration options	M
		id		String (16)	Brand ID, assigned by Licensee. This parameter must be used to identify which Licensee business unit has made a user authentication request. Please note that sending the brand ID does not allow the data separation by brand ID in Evolution Live casino Backoffice	M
		skin		String (2)	Skin ID, assigned by Evolution. Selects skin to use for the particular session, skin defines branding and table sets. Licensee can have several Live casino skins for one and the same casino	M

					instance	
	game			Object	Object containing game launch configuration and direct table launch options	O
		category		String	<p>Specifies game category what needs to be launched:</p> <ul style="list-style-type: none"> • roulette • blackjack • baccarat (can include dragontiger) • holdem • uth • eth • csp • tcp • trp • moneywheel • thb • dragontiger • topcard 	M
		interface		String	<p>Specifies the game interface version:</p> <ul style="list-style-type: none"> • view1 - launch game in 3D view • view2 - launch game in classic view • inlinevideo - launch mobile game on iPhone with inLine-Video enabled in wrapped version • hd1 - to be used for launch of csp <p>Must be used in conjunction with `config.game.table` object</p>	O
		table		Object	Object containing table details for direct game launch	O

			id	String	<p>Specifies table ID for the game table which needs to be launched.</p> <ul style="list-style-type: none"> • Either table ID or virtual table ID has to be specified for direct table launch • List of table ID's for generic tables for UAT and Live environments can be found in document "Integration manual". Virtual table IDs are different for each casino and should be requested separately 	M
			seat	Integer	Number of a seat for BlackJack tables. Array of integers must be passed to take more than one seat	O
	channel			Object	Object containing game channel configuration options	M
		wrapped		Boolean	Specifies if client is wrapped or standalone. Should only be sent as 'true' for standalone native or mobile apps.	M
		mobile		Boolean	Specifies if game is launched using mobile device	O
	urls			Object	Object containing custom URL's to be used on a game client for requested game launch. Non-encoded URL values must be used, as example: <code>"cashier": "http://</code>	O

					www.chs.ee/"`.	
		cashier		String	Value for Cashier URL in games application, assigned by Licensee	O
		responsibleGaming		String	Value for Responsible Gaming URL in games application, assigned by Licensee	O
		lobby		String	Value for Lobby/Home URL in games application; only for mobile UI, assigned by Licensee	O
		sessionTimeout		String	Value for Session Timeout URL in games application, assigned by Licensee	O
		gameHistory		String	Value for Game History URL in games application, assigned by Licensee. Used in case of Unified Reality Check solution	O
		realityCheckURL		String	Value for Reality Check Settings URL in games application, assigned by Licensee. Used in case of Unified Reality Check solution	O

1.4 User Authentication API 2.0 response format

1.4.1 Successful response

In case of successful initialization call response, User Authentication 2.0 API will return `HTTP status 200` and following response body:

```
{
  "entry": "/entry?
params=c2l0ZT1fX2RlZmF1bHRfXwpmYW1lPWVhbnRlYmQpBVVVRlX1RPS0VOPTNIYmQ5NWY1NWUwOTQy
NmRiYmFjOTcxNmNiNzEwMWE0MGMz
YTlhMjA&JSESSIONID=3ebd95f55a09426dbbac9716cb7101a40c3a9a20",
  "entryEmbedded": "/entry?
params=c2l0ZT1fX2RlZmF1bHRfXwpmYW1lPWVhbnRlYmQpBVVVRlX1RPS0VOPTNIYmQ5NWY1NWUwOTQy
NmRiYmFjOTcxNmNiNzEwMWE0MGMz
```



```
YTIhMjA&JSESSIONID=3ebd95f55a09426dbbac9716cb7101a40c3a9a20&embedded"
}
```

Response attributes:

Name	Type	Description
entry	String	Contains the URL to be added to host name and used for user redirection
entryEmbedded	String	Contains the URL to be added to host name and used for user redirection

In order to create game launch URL for the player, returned `entry` or `entryEmbedded` attribute value should be combined with according host name:

```
https://<<hostname>><<entry>>
or
https://<<hostname>><<entryEmbedded>>
```

Please note that Evolution keeps the right to change content and format of returned URLs in the values of parameters `entry` and `entryEmbedded`.

1.4.2 Failure response

In case of failure, User Authentication 2.0 API will return `HTTP status 4XX` or `HTTP status 5XX` and following response body:

```
{
  "errors": [{
    "code": "G.0",
    "message": "Could not authenticate, please review sent data and try again. If problem persists, contact customer support "
  }]
}
```

Response attributes:

Name	Type	Description
errors	Array	Array of error objects that contain failure/error codes and optional message
code	String	Error code
message	String	Optional message of failure reason

1.5 User Authentication 2.0 API error handling

Error codes are classified into following categories:

- G - generic failures
- V - validation of input parameters failed

1.5.1 Generic failures

Code	Text	Description
G.0	Could not authenticate, please review sent data and try again. If problem persists, contact customer support	System error, should be retried, in case of constant occurrences should be reported to Evolution.
G.1	Unknown casino \$casinoKey	\$casinoKey will be provided by Evolution
G.2	Provided \$apiToken for casino \$casinoKey is incorrect	\$apiToken will be provided by Evolution
G.3	Player session creation is not configured for casino \$casinoKey	\$apiToken have not been configured on Evolution side
G.4	Unable to issue token	System error, should be retried, in case of constant occurrences should be reported to Evolution
G.5	Unable to authenticate user	
G.6	Unable to create user	
G.7	Unable to save player data	
G.8	Unable to authenticate user due to: \$status	Most likely client system returned invalid \$status
G.9	Clients IP address have been rejected	Provided to Evolution client IP address for white listing is incorrect.

1.5.2 Validation failures

Code	Text
V.1	'player.update' is a boolean and accepts only true/false
V.2	'firstName' length must be at least 1 character long
V.3	'firstName' length must be no more than 50 characters
V.4	'lastName' length must be at least 1 character long
V.5	'lastName' length must be no more than 50 characters
V.6	'nickname' length must be at least 2 characters long
V.7	'nickname' length must be no more than 35 characters
V.8	'player.session' is missing
V.9	'player.session.ip' is not correct. Should be 0-255.0-255.0-255.0-255
V.10	'player.session.id' is empty or missing
V.11	'player.country' is missing
V.12	'player.currency' is missing
V.13	'player.language' is missing
V.16	uuid can't be empty
V.17	'player' is empty or missing
V.18	'config' is empty or missing
V.19	'config.brand' is missing or empty
V.20	'config.channel' is missing or empty
V.21	'config.brand.id' is empty
V.22	'config.brand.id' too long
V.23	'config.brand.skin' is empty
V.24	'config.brand.skin' too long
V.25	'config.game.type' or 'config.game.category' must be defined
V.26	'config.game.table.id' is missing or empty
V.27	'config.channel.wrapped' is missing or empty
V.28	Provided 'config.game.type' is not supported
V.29	Provided 'config.urls.\$urlType' is not valid. Should start with http://, https://, (see page 8) native://, app://
V.30	'player.nickname' \$nickname already exists
V.31	'player.country' \$country is not valid

V.32	'player.currency' \$currency is not valid
V.33	'player.currency' \$currency can not be updated for player from FT casino
V.34	Table with id \$tableId does not exist
V.35	'player.firstName' is missing
V.36	'player.lastName' is missing
V.37	'config.brand.id' must be a string
V.38	'config.brand.skin' must be a string
V.39	'config.game.category' does not exist
V.40	'config.channel.mobile' is not boolean
V.41	'player.id' is missing
V.42	'player.nickname' is missing - does`'t present now

One Wallet Integration

2 Overview

For One Wallet integration, Evolution system accesses the player's wallet in the licensee's system in real time (expected call processing time: <2sec) to retrieve player's balance and to perform credit, debit and cancel transactions. This communication is handled by the Evolution One Wallet server.

- RESTful service with lightweight JSON-formatted requests/responses
- Requests are made via HTTP POST method, containing JSON request object in request body (all responses should contain header "content-type: application/json")
- Served over HTTPS - to ensure security and data privacy, unencrypted HTTP is not supported
- Requests are sent out asynchronously. For example, for multi-step games settlement of 1st and 3rd bet can be sent out before late bet cancel of 2nd bet. Another example, Blackjack multi seat bets will be sent out for all seats as soon as betting time over.
- All service calls will pass authentication token (API token) in query parameter "authToken" - it's a string, issued/generated per integration to add authentication/validation to service calls.
- API token is configured and used in all service calls
- API token value is URL encoded

2.1 One Wallet API methods

Assuming that REST service is deployed on URL <https://my.service.host.com/api/> and authentication token value ("authToken" parameter) is "s3cr3tV4lu3" following API methods are accessible:

- **check:** <https://my.service.host.com/api/check?authToken=s3cr3tV4lu3> (CheckUserRequest / CheckUserResponse)
- **balance:** <https://my.service.host.com/api/balance?authToken=s3cr3tV4lu3> (BalanceRequest / StandardResponse)
- **debit:** <https://my.service.host.com/api/debit?authToken=s3cr3tV4lu3> (DebitRequest / StandardResponse)
- **credit:** <https://my.service.host.com/api/credit?authToken=s3cr3tV4lu3> (CreditRequest / StandardResponse)
- **cancel:** <https://my.service.host.com/api/cancel?authToken=s3cr3tV4lu3> (CancelRequest / StandardResponse)

For testing purposes following service should be implemented on test environments

- **sid:** <https://my.service.host.com/api/sid?authToken=s3cr3tV4lu3> (CheckUserRequest / CheckUserResponse)

2.2 One Wallet API request parameters

All request and response objects contains "uuid" field which represents unique identifier for each message. There should not be any additional handling based on this field and should be used only for informative purposes (e.g logging, tracing, etc)

Request	Attribute	Type	Attribute description
<i>CheckUserRequest</i>	userId	string	Player's ID which is sent by Licensee in UserAuthentication call (player.id)
	sid	string	Player's session ID which is sent by Licensee in UserAuthentication call (session.id).
	channel	object	Object containing channel details
	channel.type	string	Channel type for player in Evolution Live Casino. <ul style="list-style-type: none">• By default "M" for mobile clients, "P" for all other.
	uuid	string	Unique request id, that identifies CheckUserRequest
<i>BalanceRequest</i>	sid	string	Player's session ID
	userId	string	Player's ID, assigned by Licensee
	currency	string	Currency code (ISO 4217 3 letter code)
	game	object	Object containing game details <ul style="list-style-type: none">• In case of non-game related balance request (e.g user enters lobby) this object will be empty or null• Could be used to apply limits for specific game data, e.g limit by game.type
	game.type	string	The game type value (e.g. "blackjack", "roulette")
	game.details	object	Object containing additional game details
	game.details.table	object	Object containing table details for the game
	game.details.table.id	string	string Unique table identifier
	game.details.table.vid	string	Unique virtual table identifier (can be null in case there is no virtual table id)
	uuid	string	Unique request id, that identifies BalanceRequest
<i>DebitRequest</i>	sid	string	Player's session ID
	userId	string	Player's ID, assigned by Licensee
	currency	string	Currency code (ISO 4217 3 letter code)
	game	object	Object containing game details
	game.id	string	Unique game round id in Evolution Live Casino <ul style="list-style-type: none">• Only provided with DebitRequest/ CreditRequest/CancelRequest, not provided with BalanceRequest
	game.type	string	The game type value (e.g. "blackjack", "roulette")
	game.details	object	Object containing additional game round details
	game.details.table	object	Object containing table details for the game Round
	game.details.table.id	string	Unique table identifier

	game.details.table.vid	string	Unique virtual table identifier (can be null in case there is no virtual table id)
	transaction	object	Object containing transaction details
	transaction.id	string	The unique identifier of transaction (e.g: used to avoid duplicate bets and other validations)
	transaction.refId	string	Reference identifier for transaction, to be able to link (correlate) and/or validate credit/cancel requests to appropriate debit request
	transaction.amount	decimal	Amount of transaction, rounded to 2 decimal Symbols
	uuid	string	Unique request id, that identifies DebitRequest
<i>CreditRequest</i>	sid	string	Player's session ID
	userId	string	Player's ID, assigned by Licensee
	currency	string	Currency code (ISO 4217 3 letter code)
	game	object	Object containing game details
	game.id	string	Unique game round id in Evolution Live Casino • only provided with DebitRequest/ CreditRequest/CancelRequest, not provided with BalanceRequest
	game.type	string	The game type value (e.g. "blackjack", "holdem", "roulette", etc)
	game.details	object	Object containing additional game round details
	game.details.table	object	Object containing table details for the game Round
	game.details.table.id	string	Unique table identifier
	game.details.table.vid	string	Unique virtual table identifier (can be null in case there is no virtual table id)
	transaction	object	Object containing transaction details
	transaction.id	string	The unique identifier of transaction (e.g: used to avoid duplicate bets and other validations)
	transaction.refId	string	Reference identifier for transaction, to be able to link (correlate) and/or validate credit/cancel requests to appropriate debit request
	transaction.amount	decimal	Amount of transaction, rounded to 2 decimal Symbols
	uuid	string	Unique request id, that identifies CreditRequest
CancelRequest	Sid	string	Player's session ID
	userId	string	Players ID, assigned by Licensee
	currency	string	Currency code (ISO 4217 3 letter code)
	game	object	Object containing game details
	game.id	string	Unique game round id in Evolution Live Casino • only provided with DebitRequest/ CreditRequest/CancelRequest, not provided with BalanceRequest
	game.type	string	The game type value (e.g. "blackjack", "roulette", etc)
	game.details	object	Object containing additional game round details
	game.details.table	object	Object containing table details for the game Round
	game.details.table.id	string	Unique table identifier
	game.details.table.vid	string	Unique virtual table identifier (can be empty or

			null in case there is no virtual table id)
	transaction	object	Object containing transaction details
	transaction	string	<p>The unique identifier of transaction (e.g: used to avoid duplicate bets and other validations)</p> <p><i>* `transaction.id` will contain a transaction ID which needs to be canceled.</i></p> <p>Example: <i>debit was made with `transaction.id` = abc12345</i> <i>cancel request will contain `transaction.id` = abc12345</i></p> <p>Note on expected behavior: <i>transaction amount should not be taken when processing cancel, it could be used for additional validation only.</i> <i>Cancellation should be executed using `transaction.id` only.</i></p>
	transaction.refId	string	Reference identifier for transaction, to be able to link (correlate) and/or validate credit/cancel requests to appropriate debit request
	transaction.amount	decimal	Amount of transaction, rounded to 2 decimal Symbols
	uuid	string	Unique request id, that identifies CancelRequest

2.3 One Wallet API response parameters

Response	Attribute	Type	Attribute description
<i>CheckUserResponse</i>	status	string	Describes status of request. <ul style="list-style-type: none"> • One of the "status" enumerated values (see Appendix "Status types") • If response header is not HTTP 200, it is mapped to TEMPORARY_ERROR • If response cannot be parsed, it is mapped to TEMPORARY_ERROR • Any values that are not in the list are mapped to UNKNOWN_ERROR
	sid	string	Player's session ID which will be used in all further API service calls. <ul style="list-style-type: none"> • If null, then one that is passed in UserAuthentication call will be used for all further calls. • If empty, empty value
	uuid	string	Unique response id, that identifies <i>CheckUserResponse</i>
<i>StandardResponse</i>	status	string	Describes status of request. <ul style="list-style-type: none"> • One of the "status" enumerated values (see Appendix "Status types") • If response header is not HTTP 200, it is mapped to TEMPORARY_ERROR • If response cannot be parsed, it is mapped to TEMPORARY_ERROR • Any values that are not in the list are mapped to UNKNOWN_ERROR
	balance	decimal	Player's balance value (real money, excluding bonus). Should be filled out with player's balance, however can be empty in the following cases: <p>a) as a response to requests other than BalanceRequest if the licensee cannot reasonably obtain an accurate user balance in response to the request,</p> <p>b) in case balance cannot be obtained due to error/ failed action. Balance returned should have a precision of 2 decimal digits. Evolution does not round returned value, only uses 2 decimal values.</p>
	bonus	decimal	Player's bonus balance to be added to real balance in "balance" property, and used as total allowed bonus for user <ul style="list-style-type: none"> • Optional field <p><i>* Example:</i> <i>if response is</i> <pre>{ "status": "OK", "balance": 999.00, "bonus": 1.00, "retransmission": false }</pre> </p>

			} <i>then total available balance for games is "balance"+"bonus", in this case 999.00 + 1.00 = 1000.00</i>
	retransmission	boolean	"true" if response is a retransmission of original response (e.g if request is retried due to network failure and an original response needs to be received with "retransmission" = true). In all other cases this should be 'false' or not included in response completely. • Optional field
	uuid	string	Unique response id, that identifies <i>StandardResponse</i>

2.4 One Wallet API request/response examples

For all examples authentication token ("authToken" parameter) value is "s3cr3tV4lu3"

API method	Check
Description	Should be used for additional validation of redirected user and sid.
URL	https://my.service.host.com/api/check?authToken=s3cr3tV4lu3
Request	<pre>CheckUserRequest { "sid": "sid-parameter-from-UserAuthentication-call", "userId": "euID-parameter-from-UserAuthentication-call", "channel": { "type": "P" }, "uuid": "ce186440-ed92-11e3-ac10-0800200c9a66" }</pre>
Response	<pre>CheckUserResponse { "status": "OK", "sid": "new-sid-to-be-used-for-api-calls-qwerty", "uuid": "ce186440-ed92-11e3-ac10-0800200c9a66" }</pre>

API method	Sid
Description	Used for sid generation for REST API test automation (without dependency on test UI availability). Returned "sid" will be used for authentication in "check" * Should be only available on test environments
URL	https://my.service.host.com/api/sid?authToken=s3cr3tV4lu3
Request	<pre>CheckUserRequest { "sid": "sid-parameter-from-UserAuthentication-call", "userId": "euID-parameter-from-UserAuthentication-call", "channel": { "type": "P" }, "uuid": "ce186440-ed92-11e3-ac10-0800200c9a66" }</pre>
Response	<pre>CheckUserResponse { "status": "OK", "sid": "new-sid-to-be-used-for-api-calls-qwerty", "uuid": "ce186440-ed92-11e3-ac10-0800200c9a66" }</pre>

API method	Balance
Description	Used to get user's balance
URL	https://my.service.host.com/api/balance?authToken=s3cr3tV4lu3
Request	<p>1) BalanceRequest (lobby, or table unknown)</p> <pre>{ "sid": "sid-parameter-from-UserAuthentication-call", "userId": "euID-parameter-from-UserAuthentication-call", "game": null, "currency": "EUR", "uuid": "ce186440-ed92-11e3-ac10-0800200c9a66" }</pre> <p>2) BalanceRequest (from known table)</p> <pre>{ "sid": "sid-parameter-from-UserAuthentication-call", "userId": "euID-parameter-from-UserAuthentication-call", "game": { "type": "blackjack", "details": { "table": { "id": "aaabbbcccddeee111", "vid": "aaabbbcccddeee111" } } }, "currency": "EUR", "uuid": "ce186440-ed92-11e3-ac10-0800200c9a66" }</pre>
Response	<p>StandardResponse</p> <pre>{ "status": "OK", "balance": 999.00, "bonus": 0.00, "uuid": "ce186440-ed92-11e3-ac10-0800200c9a66" }</pre>

API method	Debit
Description	Used to debit from account (place bets)
URL	https://my.service.host.com/api/debit?authToken=s3cr3tV4lu3
Request	<p>DebitRequest</p> <pre>{ "sid": "sid-parameter-from-UserAuthentication-call", "userId": "euID-parameter-from-UserAuthentication-call", "currency": "EUR", "game": { "id": "7kfwqku4jb4mtas1n4k4irqa", "type": "blackjack", "details": {</pre>

	<pre> "table" : { "id" : "aaabbbccdddeee111", "vid" : "aaabbbccdddeee111" } }, "transaction":{ "id":"D1459zzz", "refId":"1459zzz", "amount":1.55 }, "uuid":"ce186440-ed92-11e3-ac10-0800200c9a66" } </pre>
Response	<p>StandardResponse</p> <pre> { "status":"OK", "balance":999.00, "bonus":1.00, "uuid":"ce186440-ed92-11e3-ac10-0800200c9a66" } </pre>

API method	Credit
Description	<p>Used to credit user's account (settle bets)</p> <p>* "sid" validation should be optional or accept empty sids. This may be needed some cases:</p> <ul style="list-style-type: none"> • game result was corrupted and money need to be transferred to user • when user logged out but game continues and settlement needs to be done even if user went offline • other environmental issues (e.g service was down / network is down, etc)
URL	https://my.service.host.com/api/credit?authToken=s3cr3tV4lu3
Request	<p>CreditRequest</p> <p>* Example (in case of win): CreditRequest</p> <pre> { "sid":"sid-parameter-from-UserAuthentication-call", "userId":"euID-parameter-from-UserAuthentication-call", "currency":"EUR", "game":{ "id":"7kfwqku4jb4mtas1n4k4irqa", "type":"blackjack", "details" : { "table" : { "id" : "aaabbbccdddeee111", "vid" : "aaabbbccdddeee111" } } }, "transaction":{ "id":"C1459zzz", </pre>

	<pre> "refId":"1459zzz", "amount":1.55 }, "uuid":"ce186440-ed92-11e3-ac10-0800200c9a66" } * Example (in case of loose): CreditRequest { "sid":"sid-parameter-from-UserAuthentication-call", "userId":"euID-parameter-from-UserAuthentication-call", "currency":"EUR", "game":{ "id":"7kfwqku4jb4mtas1n4k4irqa", "type":"blackjack", "details":{ "table":{ "id": "aaabbbccdddeee111", "vid": "aaabbbccdddeee111" } } }, "transaction":{ "id":"C1459zzz", "refId":"1459zzz", "amount":0.00 }, "uuid":"ce186440-ed92-11e3-ac10-0800200c9a66" } </pre>
Response	<p>StandardResponse</p> <pre> { "status":"OK", "balance":999.00, "bonus":1.00, "uuid":"ce186440-ed92-11e3-ac10-0800200c9a66" } </pre>

API method	Cancel
Description	<p>Use to cancel bets (e.g late bet)</p> <p>*"sid" validation should be optional or accept empty sids.This may be needed some cases:</p> <ul style="list-style-type: none"> • game result was corrupted and money need to be transferred to user; • when user logged out but game continues and settlement needs to be done even if user went offline; • other environmental issues (e.g service was down / network is down, etc)
URL	https://my.service.host.com/api/cancel?authToken=s3cr3tV4lu3
Request	<p>CancelRequest</p> <pre> { "sid":"sid-parameter-from-UserAuthentication-call", "userId":"euID-parameter-from-UserAuthentication-call", </pre>

	<pre> "currency": "EUR", "game": { "id": "7kfwqku4jb4mtas1n4k4irqa", "type": "blackjack", "details": { "table": { "id": "aaabbbccdddeee111", "vid": "aaabbbccdddeee111" } } }, "transaction": { "id": "D1459zzz", "refId": "1459zzz", "amount": 1.55 }, "uuid": "ce186440-ed92-11e3-ac10-0800200c9a66" </pre>
Response	<pre> StandardResponse { "status": "OK", "balance": 999.00, "bonus": 1.00, "uuid": "ce186440-ed92-11e3-ac10-0800200c9a66" } </pre>

2.5 One Wallet settlement types

A settlement type is a configurable option per integration. One Wallet settlement method is "Gamewise" and it's preferable to agree on the specific settlement types before the actual integration is started.

Gamewise settlement:

- It does not matter how many Debit request (transactions) are in the game round, there will be single Credit request at the end of the game, having aggregated amounts to be transferred to the users' wallet
- Gamewise settlement type is best suit when Licensee would like track game round state (Licensees will know that game ended, when single aggregated Credit call is being issued)
- It is expected that once a Credit request is accepted, the game (game round) becomes completely closed and any other further Credit request for the same game **should not be accepted and should not affect player's balance** regardless if the second Credit request comes with the same transaction id or with the new one.

Note, that in both cases (either Mixed or Gamewise), if Debit requests are late for the game, there will be no Credit request issued for those, only Cancel request for **each transaction**.

2.6 One Wallet transaction re-try policy

Each call for a bet or settlement/cancellation is assigned with a transaction identifier to maintain consistency and integrity of all financial transactions from-to One Wallet. As server-to-server requests may time out or respond with an error, transaction retry mechanism is being used to get the state of transaction (bet or settlement).

Request timeout value, number of retries and delays between each retry attempt is configurable in One Wallet. Default re-try configuration in OneWallet is: 5 retries with 1 min delay in between -> 10 retries with 5 min delay in between -> 24 retries with 10 min delay in between.

Retries are triggered when Evolution fails to get a successful response from server and/or transaction state is not known. Retries are processed in background and do not affect user experience and try to resolve status of transaction. Retries are persistent, therefore they are not lost and have their final state available (all failed or expired retries are stored in a FAILED or EXPIRED state). One Wallet provides a retry mechanism on following operations:

- Debit
- Credit
- Cancel

Following operations are **not** retried:

- Balance
- Check

General logic for retries:

1. retries are executed until Licensee service returns a response with status for transaction:

- Debit retries
 - if Licensee responds with status "OK" or "BET_ALREADY_EXIST", a rollback will be requested (cancellation is executed to return the funds to the user, executed in background via retry cancel queue)
 - if Licensee responds with an error code for the bet, which is not recoverable (e.g "INSUFFICIENT_FUNDS" or "INVALID_TOKEN_ID"), One Wallet will assume that transaction didn't happen on Licensee side and cancellation will not be executed
- Credit (or Cancel) retries:
 - if Licensee responds with status "OK" or "BET_ALREADY_EXIST", Credit retry will be considered as successful and retries will be stopped
 - if Licensee responds with a non-retryable error (e.g "ACCOUNT_LOCKED" or "INVALID_TOKEN_ID"), retry will be considered as failed.

2. failed or expired retries will be updated with FAILED or EXPIRED state on Evolution side and retry mechanism for the particular operation will stop. Error codes for retryable and non-retryable errors can be found in Appendix "Error Code Table" and "Status types".

Player History services

3.1 Security

3.1.1 Basics

All communication between Evolution Gaming servers and Licensee servers **must** be performed via encrypted channels using HTTPS protocol. Plain connections are not accepted.

3.1.2 Authentication

Requests to the Game History API are protected with HTTP Basic authentication ([RFC 2617 - IETF](#)) over HTTPS. Existing UserAuthentication 2.0 account `casino.key` and Game History `apiToken` have to be sent in the Authorization header as username and password, respectively.

Example:

```
Authorization: Basic dGVzdC1jYXNpbm8ua2V5OnRlc3QtYXBpVG9rZW4=
```

This header carries *Base64-encoded* pair of `test-casino.key:test-apiToken` as `username:password`

Response codes

If client omits aforementioned authentication header or provided values are not valid, the request is rejected with appropriate HTTP response code: 401 Unauthorized

3.2 Base URI

All URLs referenced in this documentation have the following base URI:
`https://«licensee_hostname»/api/gamehistory/v1`

Report type	Method	Protocol	URL
Retrieve casino daily report within specified data range	GET	https	/casino/daily-report
Retrieve game details for a casino within specified date range	GET	https	/casino/games
Retrieve game details for a casino within specified date range (JSON Streaming service)	GET	https	/casino/games/stream
Retrieve game details of a particular player's game round	GET	https	/players/{playerId}/games/{gameId}

3.3 Retrieve casino daily report within specified data range (/casino/daily-report)

"/casino/daily-report" resource URI has following limitations:

- Reporting period is restricted to 30 days maximum. `startDate` and `endDate` parameters can still be used to specify period which is less than 30 days.
- Values of `wager` and `payout` are converted to default casino currency according to currency rates known to Evolution at the time when game snapshot was generated.

Request example:

Method	URL
GET	/casino/daily-report?gameType=roulette

Resource properties

Property		Description
date		The specified date for report, (date only, no time)
gameType		The game type value for particular table
gameSubType		The game sub-type value for particular Applicable to Roulette and Baccarat games (Expected values: 'double_ball_roulette', 'french', 'lightning', 'nocommision')
table		<i>table details</i>
	id	The unique (internal) Table identifier

	name	The entity name for particular table
roundCount		The sum of all rounds played for specified day aggregated by particular table
currency		Default casino currency
wager		The sum of bet amount in all rounds played for specified day aggregated by particular table (in default casino currency)
	playerCurrency	Player's currency <ul style="list-style-type: none"> applicable if query parameter <code>playerCurrency=true</code> is being used
	amount	The sum of bet amount in all rounds played for specified day aggregated by particular table (in players currency) <ul style="list-style-type: none"> applicable if query parameter <code>playerCurrency=true</code> is being used
payout		The sum of withdrawal amount in all rounds played for specified day aggregated by particular table (in default casino currency)
	playerCurrency	Player's currency <ul style="list-style-type: none"> applicable if query parameter <code>playerCurrency=true</code> is being used
	amount	The sum of withdrawal amount in all rounds played for specified day aggregated by particular table (in players currency) <ul style="list-style-type: none"> applicable if query parameter <code>playerCurrency=true</code> is being used

JSON schema for /casino/daily-report

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "properties": {
    "data": {
      "items": {
        "properties": {
          "currency": {
            "description": "Default casino currency",
            "type": "string"
          },
          "date": {
```

```

        "description": "The specified date for report, (date only, no time)",
        "type": "string"
    },
    "payout": {
        "description": "The sum of withdrawal amount in all rounds played for specified day
aggregated by particular table (in default casino currency)",
        "type": "number"
    },
    "roundCount": {
        "description": "The sum of all rounds played for specified day aggregated by particular
table",
        "type": "integer"
    },
    "table": {
        "properties": {
            "id": {
                "description": "The unique (internal) Table identifier.",
                "type": "string",
                "example": "jhtj4343jkhtr33344"
            },
            "name": {
                "description": "The entity name for particular table",
                "type": "string",
                "example": "Immersive Roulette"
            }
        },
        "required": [
            "id",
            "name"
        ],
        "type": "object"
    },
    "gameType": {
        "description": "The game type value for particular table",
        "type": "string",
        "enum": ["roulette", "blackjack", "baccarat", "holdem", "uth", "eth", "csp", "tcp", "trp",
"moneywheel", "americanroulette", "thb", "rng-roulette", "dragontiger", "topcard", "rng-blackjack"]
    },
    "wager": {
        "description": "The sum of bet amount in all rounds played for specified day aggregated
by particular table (in default casino currency)",
        "type": "integer"
    }
},
"required": [
    "payout",
    "roundCount",

```

```

        "currency",
        "date",
        "table",
        "wager",
        "gameType"
    ],
    "type": "object"
},
"type": "array"
},
"timestamp": {
    "description": "Time and date when response was generated.",
    "type": "string"
},
"uuid": {
    "description": "Universally unique identifier of the request.",
    "type": "string"
}
},
"required": [
    "data",
    "timestamp",
    "uuid"
],
"type": "object"
}

```

Query parameters:

Name	Type	Description	Mandatory/Optional
startDate	string (UTC date)	Specifies the starting date of the report time range (in the format “YYYY-MM-DD”), inclusive. Example: 2017-02-22. Default: today	O
endDate	string (UTC date)	Specifies the end date of the report time range (in the format “YYYY-MM-DD”), inclusive. If endDate is specified – then startDate is mandatory. Default: none, report will contain daily summary until the current moment.	O

gameType	string	<p>Specifies valid game types:</p> <ul style="list-style-type: none"> • roulette • blackjack • baccarat • holdem • uth • eth • csp • tcp • trp • moneywheel • americanroulette • thb • rng-roulette • dragontiger • topcard • topcard • rng-blackjack 	O
channel	string	<p>Specifies channel type for the report:</p> <ul style="list-style-type: none"> • desktop • mobile (tablets and mobile phones in included) • other (devices which are not recognized by device recognition library) <p><i>If not specified, game history data is accumulated for all channels: desktop + mobile + other</i></p>	O
playerCurrency	boolean	<p>Specifies if wager and payout amounts shall be represented in player's currency. Possible values: true or false. Default: false</p>	O

3.4 Retrieve game details for a casino within specified date range (/casino/games)

"/casino/games" resource URI has following limitations:

- Reporting period is restricted to 24 hours maximum. **StartDate** and **endDate** parameters can be used to specify period which is less than 24 hours.

- Service can be periodically invoked to retrieve new game rounds information by specifying query parameter `startDate=time_of_last_invocation - 5 minutes` (without specifying `endDate`).
- API responses in subsequent invocations may contain data entries already present in previous invocations (duplicates) and it is up to the licensee to handle such entries accordingly.
- Values of `wager` and `payout` are converted to default casino currency according to currency rates known to Evolution at the time when game snapshot was generated.
- In cases where the player has multiple sessions during the game round (for example, if user disconnected and reconnected while game round was in progress), `sessionId` and `casinoSessionId` fields will contain values from one of the sessions of the round.

Request example:

Method	URL
GET	<code>/casino/games?startDate=2017-01-23T10:02:59.117Z</code>

Resource properties

Property				Description
date				The specified date for report, (date only, no time)
games				<i>Array of game rounds</i>
	id			The unique (internal) game round identifier
	startedAt			The date and time when game round started

	settledAt			The date and time when game round settled
	status			Game round status (Resolved/Cancelled)
	gameType			The game type value for particular table
	gameSubType			The game sub-type value for particular Applicable to Roulette and Baccarat games (Expected values: 'double_ball_roulette', 'french', 'lightning', 'nocommission')
	table			<i>Table details</i>
		id		The unique (internal) Table identifier
		name		The entity name for particular table
	dealer			<i>Dealer details</i>
		uid		The unique (internal) dealer identifier
		name		Dealer's name
	currency			Default casino currency
	wager			The sum of players total bet amount in particular game round
	payout			The sum of players total withdrawal amount in particular game round
	participants			<i>Provides details of every player who participated in particular game round</i>
		casinoId		Casino unique identifier, assigned by Evolution
		playerId		Unique external (assigned by Licensee) user ID
		screenName		Player's alias (nickname)
		sessionId		Player's session ID, assigned by Evolution
		casinoSessionId		Player's session ID, assigned

				by Licensee
		currency		Currency code (ISO 4217 3 letter code)
			betCoverage.simple	Represents probability to win for a specific player. Values in range [0, 1] <ul style="list-style-type: none"> applicable to all Roulette types except Double Ball Roulette
		bets		<i>Array of player bet details</i>
			code	Game specific bet code (internal)
			stake	The bet amount what player placed in particular game round
			payout	The withdrawal amount what player received in particular game round
			placedOn	The date and time when bet was placed
			description	Human-readable bet description
			transactionId	The unique identifier of external transaction: <ul style="list-style-type: none"> Games (e.g. Roulette and Baccarat) where all bets has to be placed in the begining of the game round are using same external transactionId for each player stake. Games (e.g. Blackjack

				<p>and Poker) which allow bets to be placed at different stages of the game round are using different external transactionId for each player stake depending on the stage of the game round.</p>
		configOverlays		Array of configuration overlay id's (internal)
		seats		<p>Player results on one or more seats (including split seats and bet behind - not including decisions because these belong to seats/pure game)</p> <ul style="list-style-type: none"> • applicable for BlackJack
		decisions		<p>Player decisions during the game round</p> <ul style="list-style-type: none"> • applicable for Poker games
		sideBetPlayerPair		Result for Baccarat Player Pair bonus side-bet
		sideBetBankerPair		Result for Baccarat Banker Pair bonus side-bet
		sideBetPerfectPair		Result for Baccarat Perfect

				Pair bonus side-bet
		sideBetEitherPair		Result for Baccarat Either Pair bonus side-bet
		sideBetPlayerBonus		Result for Baccarat Player Bonus bonus side-bet
		sideBetBankerBonus		Result for Baccarat Banker Bonus bonus side-bet
		sideBetSuperSix		Result for Baccarat Super 6 bonus side-bet
		sideBetAABonus		Result for Casino Holdem Bonus side-bet
		sideBet5p1		Result for Caribbean Stud Poker 5+1 bonus side-bet
		sideBetPairPlus		Result for Three Card Poker Pair Plus bonus side-bet
		sideBet6CardBonus		Result for Three Card Poker 6 Card bonus side-bet
		sideBetPairOrBetter		Result for Triple Card Poker Pair or Better bonus side-bet
		sideBetTrips		Result for Ultimate Texas Holdem Trips bonus side-bet
		sideBetBestFive		Result for Extreme Texas Holdem Best Five bonus side-bet
		suitedTie		Result for Dragon Tiger suited tie bet
	result			<i>Provides outputs from game(cards, balls, etc) and describes game outcome interpretation according to game rules, e.g., specifying winning side, providing totals, etc</i>

JSON schema for /casino/games

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "properties": {
```

```
"data": {
  "items": {
    "properties": {
      "date": {
        "description": "The date for game round, (date only, no time)",
        "type": "string",
        "example": "2017-01-03"
      },
      "games": {
        "items": {
          "properties": {
            "currency": {
              "description": "Default casino currency",
              "type": "string",
              "example": "EUR"
            },
            "dealer": {
              "properties": {
                "name": {
                  "description": "Dealer's name",
                  "type": "string"
                },
                "uid": {
                  "description": "The unique (internal) dealer identifier",
                  "type": "string"
                }
              },
              "required": [
                "uid",
                "name"
              ],
              "type": "object"
            },
            "id": {
              "description": "The unique game round identifier",
              "type": "string"
            },
            "participants": {
              "items": {
                "properties": {
                  "bets": {
                    "items": {
                      "properties": {
                        "code": {
                          "description": "Game specific bet code (internal)",
                          "type": "string"
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```

        "description": {
            "description": "Human-readable bet description",
            "type": "string"
        },
        "payout": {
            "description": "The sum of players total withdrawal amount in
particular game round",
            "type": "integer"
        },
        "placedOn": {
            "description": "The date and time when bet was placed",
            "type": "string",
            "example": "2017-02-08T13:07:40.222Z"
        },
        "stake": {
            "description": "The bet amount what player placed in particular game
round",
            "type": "integer"
        },
        "transactionId": {
            "description": "The unique identifier of external transaction",
            "type": "string"
        }
    },
    "required": [
        "payout",
        "code",
        "description",
        "stake",
        "placedOn",
        "transactionId"
    ],
    "type": "object"
},
"type": "array"
},
"casinoId": {
    "description": "Casino unique identifier, assigned by Evolution",
    "type": "string",
    "example": "thebest000000000001"
},
"configOverlays": {
    "items": {
        "description": "Array of configuration overlay id's (internal)",
        "type": "string"
    },
    "type": "array"
}

```

```

    },
    "currency": {
      "description": "Currency code (ISO 4217 3 letter code)",
      "type": "string",
      "example": "EUR"
    },
    "betCoverage": {
      "description": "Represents probability to win for a specific player. Values in
range [0, 1]. Applicable for all Roulette types except Double Ball Roulette",
      "type": "object"
    },
    "playerId": {
      "description": "Unique external (assigned by Licensee) user ID",
      "type": "string"
    },
    "screenName": {
      "description": "Player's alias (nickname)",
      "type": "string"
    },
    "sessionId": {
      "description": "Player's session ID, assigned by Evolution",
      "type": "string"
    },
    "casinoSessionId": {
      "description": "Player's session ID, assigned by Licensee",
      "type": "string"
    },
    "sideBetBankerBonus": {
      "description": "Applicable only for Baccarat. Outcome of the side bet Banker
Bonus",
      "type": "string",
      "enum": ["Win", "Lost"]
    },
    "sideBetBankerPair": {
      "description": "Applicable only for Baccarat. Outcome of the side bet Banker
Pair",
      "type": "string",
      "enum": ["Win", "Lost"]
    },
    "sideBetEitherPair": {
      "description": "Applicable only for Baccarat. Outcome of the side bet Either
Pair",
      "type": "string",
      "enum": ["Win", "Lost"]
    },
    "sideBetPerfectPair": {
      "description": "Applicable only for Baccarat. Outcome of the side bet Perfect

```

```

Pair",
    "type": "string",
    "enum": ["Win", "Lost"]
  },
  "sideBetPlayerBonus": {
    "description": "Applicable only for Baccarat. Outcome of the side bet Player
Bonus",
    "type": "string",
    "enum": ["Win", "Lost"]
  },
  "sideBetPlayerPair": {
    "description": "Applicable only for Baccarat. Outcome of the side bet Player
Pair",
    "type": "string",
    "enum": ["Win", "Lost"]
  },
  "sideBetSuperSix": {
    "description": "Applicable only for Baccarat. Outcome of the side bet Super
6",
    "type": "string",
    "enum": ["Win", "Lost"]
  },
  "suitedTie": {
    "description": "Applicable only for DragonTiger. Outcome of the Suited Tie
bet",
    "type": "string",
    "enum": ["Win", "Lose"]
  },
  "seats": {
    "type": "object",
    "description": "Applicable only for BlackJack. Player results on one or more
seats (including split seats and bet behind - not including decisions because these belong to seats/pure
game)",
    "properties": {
      "Seat_number": {
        "properties": {
          "betBehind": {
            "description": " BJ Bet Behind bet",
            "type": "boolean"
          },
          "doubleDown": {
            "description": " BJ Double down bet",
            "type": "boolean"
          },
          "insurance": {
            "description": "BJ Insurance bet",
            "type": "boolean"
          }
        }
      }
    }
  }
}

```

```

    },
    "sideBetPerfectPair": {
      "properties": {
        "betCode": {
          "description": "Human readable bet description",
          "type": "string"
        },
        "result": {
          "description": "Outcome of the side bet.",
          "type": "string",
          "enum": ["Win", "Lost"]
        }
      },
      "type": "object"
    },
    "splitHand": {
      "description": "BJ Split hand",
      "type": "boolean"
    }
  },
  "type": "object"
}
},
"sideBetAABonus": {
  "description": "Applicable only for Casino Holdem",
  "properties": {
    "betCode": {
      "description": "Human readable bet description",
      "type": "string"
    },
    "result": {
      "description": "Outcome of the side bet",
      "type": "string",
      "enum": ["Win", "Lost"]
    }
  },
  "type": "object"
},
"sideBet5p1": {
  "description": "Applicable only for Caribbean Stud Poker",
  "properties": {
    "betCode": {
      "description": "Human readable bet description",
      "type": "string"
    },
    "result": {

```



```

        "description": "Outcome of the side bet",
        "type": "string",
        "enum": ["Win", "Lost"]
    }
},
"type": "object"
},
"sideBetPairPlus": {
    "description": "Applicable only for Three Card Poker",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",
            "enum": ["Win", "Lost"]
        }
    },
    "type": "object"
},
"sideBet6CardBonus": {
    "description": "Applicable only for Three Card Poker",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",
            "enum": ["Win", "Lost"]
        }
    },
    "type": "object"
},
"sideBetPairOrBetter": {
    "description": "Applicable only for Triple Card Poker",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",

```

```
        "enum": ["Win", "Lost"]
    }
},
"type": "object"
},
"sideBetTrips": {
    "description": "Applicable only for Ultimate Texas Holdem",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",
            "enum": ["Win", "Lost"]
        }
    },
    "type": "object"
},
"sideBetBestFive": {
    "description": "Applicable only for Extreme Texas Holdem",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",
            "enum": ["Win", "Lost"]
        }
    },
    "type": "object"
},
"decisions": {
    "description": "Applicable only for Poker game",
    "items": {
        "properties": {
            "type": {
                "description": "Decision type",
                "type": "string",
                "example": "Call"
            }
        },
        "required": [
            "type"
        ]
    }
}
```

```

        ],
        "type": "object"
    },
    "type": "array"
}
},
"required": [
    "bets",
    "casinoId",
    "playerId",
    "screenName",
    "currency",
    "sessionId",
    "casinoSessionId",
    "configOverlays"
],
"type": "object"
},
"type": "array"
},
"payout": {
    "description": "The sum of players total withdrawal amount in particular game
round",
    "type": "integer"
},
"result": {
    "type": "object",
    "description": "Provides outputs from game(cards, balls, etc) and describes game
outcome interpretation according to game rules, e.g., specifying winning side, providing totals, etc.
Please see example outputs for each game type"
},
"settledAt": {
    "description": "The date and time when game round settled",
    "type": "string",
    "example": "2017-01-03T10:03:40.246Z"
},
"startedAt": {
    "description": "The date and time when game round started",
    "type": "string",
    "example": "2017-01-03T10:02:59.117Z"
},
"status": {
    "description": "Game round status",
    "type": "string",
    "enum": ["Resolved", "Cancelled"]
},
"table": {

```

```

        "properties": {
            "id": {
                "description": "The unique (internal) Table identifier",
                "type": "string"
            },
            "name": {
                "description": "The entity name for particular table",
                "type": "string",
                "example": "Roulette VIP"
            }
        },
        "required": [
            "id",
            "name"
        ],
        "type": "object"
    },
    "gameType": {
        "description": "The game type value for particular table",
        "type": "string",
        "enum": ["roulette", "blackjack", "baccarat", "holdem", "uth", "eth", "csp", "tcp",
"trp", "moneywheel", "americanroulette", "thb", "rng-roulette".
        "topcard", "dragontiger", "rng-blackjack"
    ]
    },
    "wager": {
        "description": "The sum of players total bet amount in particular game round",
        "type": "integer"
    }
},
"required": [
    "status",
    "payout",
    "settledAt",
    "dealer",
    "currency",
    "participants",
    "result",
    "table",
    "startedAt",
    "wager",
    "gameType",
    "id"
],
"type": "object"
},
"type": "array"

```

```

    }
  },
  "required": [
    "date",
    "games"
  ],
  "type": "object"
},
"type": "array"
},
"timestamp": {
  "description": "Time and date when response was generated.",
  "type": "string"
},
"uuid": {
  "description": "Universally unique identifier of the request.",
  "type": "string"
}
},
"required": [
  "data",
  "timestamp",
  "uuid"
],
"type": "object"
}

```

Query parameters:

Name	Type	Description	Mandatory/Optional
startDate	string (UTC date+time)	Specifies the starting date of the report time range (in the format “YYYY-MM-DDTHH:mm:ss.SSSZ”), inclusive. Example: 2017-02-22T13:49:59.410Z. <i>Default: beginning of current day</i>	O
endDate	string (UTC date+time)	Specifies the end date of the report time range (in the format “YYYY-MM-DDTHH:mm:ss.SSSZ”), inclusive. If endDate is specified – then startDate is mandatory. <i>Default: startDate + 24h, for current day report will contain all games</i>	O

		<i>known until the current moment</i>	
gameType	string	<p>Specifies valid game types:</p> <ul style="list-style-type: none"> • roulette • blackjack • baccarat • holdem • uth • eth • csp • tcp • trp • moneywheel • americanroulette • thb • rng-roulette • dragontiger • topcard • rng-blackjack 	O
channel	string	<p>Specifies channel type for the report:</p> <ul style="list-style-type: none"> • desktop • mobile (tablets and mobile phones included) • other (devices which are not recognized by device recognition library) <p><i>If not specified, game history data is accumulated for all channels:</i></p>	O

		<i>desktop + mobile + other</i>	
transactionId	string	<p>Specifies to search a game round by transaction ID of a bet participating in this game round.</p> <ul style="list-style-type: none"> TransactionId query parameter must be used together with startDate query parameter specifying the date when particular transaction took place. 	

3.5 Retrieve game details for a casino within specified date range (JSON-streaming service) (/casino/games/stream)

This service endpoint is an extension for casino history service endpoint /casino/games/ and it serves the purpose of delivery of large payloads where API response size may exceed 10000 game snapshot record entries.

Data is streamed using Chunked transfer encoding (for more detailed information please see RFC 7230). The body of the API response consists of line-delimited JSON game snapshots (see below response body example).

Resource properties for /casino/games/stream service endpoint are the same as for /casino/games/ service endpoint.

Request example:

Method	URL
GET	/casino/games/stream?startDate=2017-01-23T10:02:59.117Z

Response body:

```
{ "id": "a1s2d3f4g5h6j7k8", "startedAt": "2017-01-03T10:02:59.117Z", "settledAt": "2017-01-03T10:03:40.246Z", "status": "Resolved", "gameType": "roulette", "table": { "id": "a1s2d3f4g5h6j7k8",
```

```

"name": "Roulette VIP" }, "dealer": { "uid": "z1x2c3v4b5n6", "name": "Gnome Ann" }, "currency":
"EUR", "wager": 1750.00, "payout": 2000.00, "participants": [{ "casinoId": "thebest000000000001",
"playerId": "abc123456789", "screenName": "John Doe", "sessionId": "3232jh5488fj88df87f8",
"casinoSessionId": "3jh4jg43ghdf77f8", "currency": "EUR", "bets": [{
"code": "ROU_1Red", "stake": 5, "payout": 0, "placedOn": "2017-02-08T13:07:40.222Z", "description":
"1 Red", "transactionId": "1234" }, { "code": "ROU_32Red", "stake": 1, "payout": 36, "placedOn": "2017-
02-08T13:07:44.223Z", "description": "32 Red", "transactionId": "1234" }], "configOverlays":
["virtual_table_id"] }], "result": { "outcomes": [{ "number": "32", "type": "Even", "color": "Red" } ]}}
{snapshot2-json}
{snapshot3-json}
...
{snapshotN-json}

```

3.6 Retrieve game details of a particular player's game round (*/players/{playerId}/games/{gameId}*)

- In cases where the player has multiple sessions during the game round (for example, if user disconnected and reconnected while game round was in progress), sessionId and casinoSessionId fields will contain values from one of the sessions of the round.

Request example:

```
GET /players/123456789/games/za1xs2cd3vf4bg579
```

Resource properties

Property			Description
id			The unique game round identifier
startedAt			The date and time when game round started_
settledAt			The date and time when game round settled
status			Game round status (Resolved/Cancelled)
aams			<i>Optional for AAMS-regulated games</i>
	participationId		AAMS Participation Id
	gameSessionId		AAMS Session Id
gameType			The game type value for particular table

gameSubType			The game sub-type value for particular Applicable to Roulette and Baccarat games (Expected values: 'double_ball_roulette', 'french', 'lightning', 'nocommission')
table			<i>Table details</i>
	id		The unique (internal) Table identifier
	name		The entity name for particular table
dealer			<i>Dealer details</i>
	uid		The unique (internal) dealer identifier
	name		Dealer's name
result			<i>Provides outputs from game(cards, balls, etc) and describes game outcome interpretation according to game rules, e.g., specifying winning side, providing totals, etc</i>
participants			<i>Provides details of every player who participated in particular game round</i>
	casinoId		Casino unique identifier, assigned by Evolution
	playerId		Unique external (assigned by Licensee) user ID
	screenName		Player's alias (nickname)
	sessionId		Player's session ID, assigned by Evolution
	casinoSessionId		Player's session ID, assigned by Licensee
	currency		Currency code (ISO 4217 3 letter code)
	betCoverage.simple		Represents probability to win for a specific player. Values in range [0, 1] <ul style="list-style-type: none"> applicable to all Roulette types except Double Ball Roulette
	bets		<i>Array of player bet details</i>
		code	Game specific bet code (internal)
		stake	The bet amount what player placed in particular game round
		payout	The withdrawal amount what player received in particular game round
		placedOn	The date and time when bet was placed

		description	Human-readable bet description
		transactionId	<p>The unique identifier of external transaction:</p> <ul style="list-style-type: none"> Games (e.g. Roulette and Baccarat) where all bets has to be placed in the begining of the game round are using same external transactionId for each player stake. Games (e.g. Blackjack and Poker) which allow bets to be placed at different stages of the game round are using different external transactionId for each player stake depending on the stage of the game round.
	configOverlays		Array of configuration overlay id's (internal)
	seats		<p>Player results on one or more seats (including split seats and bet behind - not including decisions because these belong to seats/pure game)</p> <ul style="list-style-type: none"> applicable for BlackJack
	decisions		<p>Player decisions during the game round</p> <ul style="list-style-type: none"> applicable for Poker games
	sideBetPlayerPair		Result for Baccarat Player Pair bonus side-bet
	sideBetBankerPair		Result for Baccarat Banker Pair bonus side-bet
	sideBetPerfectPair		Result for Baccarat Perfect Pair bonus side-bet
	sideBetEitherPair		Result for Baccarat Either Pair bonus side-bet
	sideBetPlayerBonus		Result for Baccarat Player Bonus bonus side-bet
	sideBetBankerBonus		Result for Baccarat Banker Bonus bonus side-bet
	sideBetSuperSix		Result for Baccarat Super 6 bonus side-bet
	sideBetAABonus		Result for Casino Holdem Bonus side-bet
	sideBet5p1		Result for Caribbean Stud Poker 5+1 bonus side-bet

	sideBetPairPlus		Result for Three Card Poker Pair Plus bonus side-bet
	sideBet6CardBonus		Result for Three Card Poker 6 Card bonus side-bet
	sideBetPairOrBetter		Result for Triple Card Poker Pair or Better bonus side-bet
	sideBetTrips		Result for Ultimate Texas Holdem Trips bonus side-bet
	sideBetBestFive		Result for Extreme Texas Holdem Best Five bonus side-bet
	suitedTie		Result for Dragon Tiger suited tie bet

JSON schema for /players/{playerId}/games/{gameId}

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "properties": {
    "data": {
      "items": {
        "properties": {
          "aams": {
            "description": "Applicable only for AAMS-regulated games",
            "properties": {
              "gameSessionId": {
                "description": "AAMS Participation Id",
                "type": "string"
              },
              "participationId": {
                "description": "AAMS Session Id",
                "type": "string"
              }
            }
          },
          "required": [
            "participationId",
            "gameSessionId"
          ],
          "type": "object"
        },
        "dealer": {
          "properties": {
            "name": {
              "description": "Dealer's name",
              "type": "string"
            }
          },
          "type": "object"
        }
      }
    }
  }
}
```

```

        "uid": {
            "description": "The unique (internal) dealer identifier",
            "type": "string"
        },
    },
    "required": [
        "uid",
        "name"
    ],
    "type": "object"
},
"id": {
    "description": "The unique game round identifier",
    "type": "string"
},
"participants": {
    "items": {
        "properties": {
            "bets": {
                "items": {
                    "properties": {
                        "code": {
                            "description": "Game specific bet code (internal)",
                            "type": "string"
                        },
                        "description": {
                            "description": "Human-readable bet description",
                            "type": "string"
                        },
                        "payout": {
                            "description": "The sum of players total withdrawal amount in particular game
round",
                            "type": "integer"
                        },
                        "placedOn": {
                            "description": "The date and time when bet was placed",
                            "type": "string",
                            "example": "2017-02-08T13:07:40.222Z"
                        },
                        "stake": {
                            "description": "The bet amount what player placed in particular game round",
                            "type": "integer"
                        },
                        "transactionId": {
                            "description": "The unique identifier of external transaction",
                            "type": "string"
                        }
                    },
                    "required": [
                        "payout",

```

```

        "code",
        "description",
        "stake",
        "placedOn",
        "transactionId"
    ],
    "type": "object"
},
"type": "array"
},
"casinoId": {
    "description": "Casino unique identifier, assigned by Evolution",
    "type": "string",
    "example": "thebest00000000001"
},
"configOverlays": {
    "items": {
        "description": "Array of configuration overlay id's (internal)",
        "type": "string"
    },
    "type": "array"
},
"currency": {
    "description": "Currency code, (ISO 4217 3 letter code)",
    "type": "string",
    "example": "EUR"
},
"betCoverage": {
    "description": "Represents probability to win for a specific player. Values in range [0, 1].
Applicable for all Roulette types except Double Ball Roulette",
    "type": "object"
},
"playerId": {
    "description": "Unique external (assigned by Licensee) user ID",
    "type": "string"
},
"screenName": {
    "description": "Player's alias (nickname)",
    "type": "string"
},
"sessionId": {
    "description": "Player's session ID, assigned by Evolution",
    "type": "string"
},
"casinoSessionId": {
    "description": "Player's session ID, assigned by Licensee",
    "type": "string"
},
"sideBetBankerBonus": {
    "description": "Applicable only for Baccarat. Outcome of the side bet Banker Bonus",

```

```

        "type": "string",
        "enum": ["Win", "Lost"]
    },
    "sideBetBankerPair": {
        "description": "Applicable only for Baccarat. Outcome of the side bet Banker Pair",
        "type": "string",
        "enum": ["Win", "Lost"]
    },
    "sideBetEitherPair": {
        "description": "Applicable only for Baccarat. Outcome of the side bet Either Pair",
        "type": "string",
        "enum": ["Win", "Lost"]
    },
    "sideBetPerfectPair": {
        "description": "Applicable only for Baccarat. Outcome of the side bet Perfect Pair",
        "type": "string",
        "enum": ["Win", "Lost"]
    },
    "sideBetPlayerBonus": {
        "description": "Applicable only for Baccarat. Outcome of the side bet Player Bonus",
        "type": "string",
        "enum": ["Win", "Lost"]
    },
    "sideBetBankerPair": {
        "description": "Applicable only for Baccarat. Outcome of the side bet Banker Pair",
        "type": "string",
        "enum": ["Win", "Lost"]
    },
    "sideBetSuperSix": {
        "description": "Applicable only for Baccarat. Outcome of the side bet Super 6",
        "type": "string",
        "enum": ["Win", "Lost"]
    },
    "suitedTie": {
        "description": "Applicable only for Dragon Tiger. Outcome of the suited tie bet",
        "type": "string",
        "enum": ["Win", "Lost"]
    },
    "seats": {
        "type": "object",
        "description": "Applicable only for BlackJack. Player results on one or more seats
(including split seats and bet behind - not including decisions because these belong to seats/pure game)",
        "properties": {
            "Seat_number": {
                "properties": {
                    "betBehind": {
                        "description": " BJ Bet Behind bet",
                        "type": "boolean"
                    },
                    "doubleDown": {

```

```

        "description": " BJ Double down bet",
        "type": "boolean"
    },
    "insurance": {
        "description": "BJ Insurance bet",
        "type": "boolean"
    },
    "sideBetPerfectPair": {
        "properties": {
            "betCode": {
                "description": "Human readable bet description",
                "type": "string"
            },
            "result": {
                "description": "Outcome of the side bet.",
                "type": "string",
                "enum": ["Win", "Lost"]
            }
        },
        "type": "object"
    },
    "splitHand": {
        "description": "BJ Split hand",
        "type": "boolean"
    }
},
"type": "object"
}
},
"sideBetAABonus": {
    "description": "Applicable only for Casino Holdem.",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",
            "enum": ["Win", "Lost"]
        }
    },
    "type": "object"
},
"sideBet5p1": {
    "description": "Applicable only for Caribbean Stud Poker",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",

```

```

        "type": "string"
    },
    "result": {
        "description": "Outcome of the side bet",
        "type": "string",
        "enum": ["Win", "Lost"]
    }
},
"type": "object"
},
"sideBetPairPlus": {
    "description": "Applicable only for Three Card Poker",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",
            "enum": ["Win", "Lost"]
        }
    },
    "type": "object"
},
"sideBet6CardBonus": {
    "description": "Applicable only for Three Card Poker",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",
            "enum": ["Win", "Lost"]
        }
    },
    "type": "object"
},
"sideBetPairOrBetter": {
    "description": "Applicable only for Triple Card Poker",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",

```



```

        "enum": ["Win", "Lost"]
    },
    "type": "object"
},
"sideBetTrips": {
    "description": "Applicable only for Ultimate Texas Holdem",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",
            "enum": ["Win", "Lost"]
        }
    },
    "type": "object"
},
"sideBetBestFive": {
    "description": "Applicable only for Extreme Texas Holdem",
    "properties": {
        "betCode": {
            "description": "Human readable bet description",
            "type": "string"
        },
        "result": {
            "description": "Outcome of the side bet",
            "type": "string",
            "enum": ["Win", "Lost"]
        }
    },
    "type": "object"
},
"decisions": {
    "description": "Applicable only for Poker game",
    "items": {
        "properties": {
            "type": {
                "description": "Decision type",
                "type": "string",
                "example": "Call"
            }
        },
        "required": [
            "type"
        ],
        "type": "object"
    },
    "type": "object"
},

```

```

        "type": "array"
      }
    },
    "required": [
      "bets",
      "casinoId",
      "playerId",
      "screenName",
      "currency",
      "sessionId",
      "casinoSessionId",
      "configOverlays"
    ],
    "type": "object"
  },
  "type": "array"
},
"result": {
  "type": "object",
  "description": "Provides outputs from game(cards, balls, etc) and describes game outcome interpretation according to game rules, e.g., specifying winning side, providing totals, etc. Please see example outputs for each game type"
},
"settledAt": {
  "description": "The date and time when game round settled",
  "type": "string",
  "example": "2017-01-03T10:03:40.246Z"
},
"startedAt": {
  "description": "The date and time when game round started",
  "type": "string",
  "example": "2017-01-03T10:02:59.117Z"
},
"status": {
  "description": "Game round status",
  "type": "string",
  "enum": ["Resolved", "Cancelled"]
},
"table": {
  "properties": {
    "id": {
      "description": "The unique (internal) Table identifier",
      "type": "string"
    },
    "name": {
      "description": "The entity name for particular table",
      "type": "string",
      "example": "Roulette VIP"
    }
  }
},

```

```

        "required": [
            "id",
            "name"
        ],
        "type": "object"
    },
    "gameType": {
        "description": "The game type value for particular table",
        "type": "string",
        "enum": ["roulette", "blackjack", "baccarat", "holdem", "uth", "eth", "csp", "tcp", "trp",
"moneywheel", "americanroulette", "thb", "rng-roulette", "dragontiger", "topcard", "rng-blackjack"]
    }
},
    "required": [
        "status",
        "settledAt",
        "dealer",
        "participants",
        "result",
        "table",
        "startedAt",
        "gameType",
        "id"
    ],
    "type": "object"
},
    "type": "array"
},
    "timestamp": {
        "description": "Time and date when response was generated.",
        "type": "string"
    },
    "uuid": {
        "description": "Universally unique identifier of the request.",
        "type": "string"
    }
},
    "required": [
        "data",
        "timestamp",
        "uuid"
    ],
    "type": "object"
}

```