KA GAMING SYSTEMS INTEGRATION

v2.38

KA Gaming

KA Gaming

All efforts have been made to ensure the accuracy of the content of this document at the time of publication. However, this document is subject to future change. Updated versions of this document will be released as needed.

KA Gaming

Table of Contents

KA Gaming

1. GLOSSARY

Term	Definition
KAGA RMP (RMP)	KAGA platform and game server that interacts with licensee wallets and players
Licensee	Licensee of KA Gaming Systems Limited games
Wallet	Licensee's mechanism for keeping a player's balance.
Player	End customer player of game
Request	A message initiated and sent between RMP and Wallet
Response	A message responding to RMP and Wallet requests
Transaction	Signifies a game play or other player balance altering event
Launcher	KAGA's specific game launch URLs and links

2. KA GAMES MINIMUM REQUIREMENTS

All games are HTML5 and are equally playable and fun on modern mobile and PC web browsers.

Verified & Supported Browsers:

- PC and Mac (Windows XP, Windows 7, Windows 8, Windows 10, MacOS 10.5+
  - IE10+
  - Chrome (all versions since 2014)
  - Safari v7+
  - FireFox v30+
  - Qihoo 360 (all versions since 2014)
  - QQ Browser(all versions since 2014)
  - Sougou (all versions since 2014)
- iOS7+ (recommended iOS9+)
  - All Safari versions
  - Chrome
  - QQ Browser
  - Qihoo 360
- Android 4.4+ (recommended Android 5.1+)
  - Chrome
  - QQ Browser
  - Qihoo 360
- AppleTV
  - Safari (all AppleTV Safari versions)
- Android TV
  - Chrome

Game Resolution & Aspect Ratio

- 16:9 or 3:2
- The 3:2 native resolution is 960x640 and 1136x640 for 16:9. Changing the window size of the game will automatically scale the game images up or down while maintaining the same aspect ratio

Network Connectivity

- Broadband/WIFI
  - Recommended minimum 2mbps broadband Internet connectivity
  - Recommended 5mbps+ for best play experience
- Mobile data networks
  - Recommended minimum 3G mobile data network connectivity
  - Recommended 4G+ or LTE+ for best play experience

Supported Languages:

- English
- Simplified Chinese

- Traditional Chinese
- Japanese
- Russian
- Thai

KA Gaming

3. GAME LIST RETRIEVAL

The licensee should call the gamelist command on the RMP system to get the latest list of available/configured games available to the licensee. This list contains details about the game type, bonus features, game identifiers, configured max bets, configured bet levels, image URL's, and other information useful for rendering of licensee game lobby page.

The gamelist API is a REST web service hosted on KAGA RMP and can be called by the licensee. A security hash code is required as a request parameter directly on the Query URL. Please see the sections below on how the security hash is generated.

Content-type: application/json
URL: https://kagarmp_server/kaga/gameList?hash=<HMAC>
POST:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
language	Yes	String	ISO-639-1 language code. KA supports multiple languages. If the passed in language is not supported, the response will default to English ("en"). Please contact your KA sales or technical contact if you need additional languages supported. Note also that for Chinese, using "zh" alone as the language will default to Simplified Chinese. In order to get Traditional Chinese, please use "zh_TW" or "zh-Hant" as the language
randomId	Yes	Int	A random number that changes for every game list call

KAGA RMP Response (JSON format):

Parameter	Required	Type	Usage
status	Yes	String	The status message of the gameList request
statusCode	Yes	Int	Status code for the request and subsequent response
gameLaunchURL	Yes	String	The URL prefix to used when constructing game launch URL's
language	Yes	String	The ISO-639-1 code of the language returned in the results
numGames	No	Int	Number of games configured for licensee if request is valid
Games	No	Array	A list of available games and game configurations for the licensee if request is valid

The response games Array contains game objects in JSON format:

Parameter	Required	Type	Usage
gameId	Yes	String	Unique name or id of the game
gameType	Yes	String	Returns “slots”, “table”, “vpoker”, “fish”, “pvp”, or “other”
variantType	Yes	String	Returns game type variant e.g. “lines” or “ways” for slots, “blackjack”, “bacaratt”, etc…
gameName	Yes	String	The localized name of the game based on the language passed in from the request
numReels	No	Int	Number of reels if it is a slots game
numRows	No	Int	Number of rows if it is a slots game
numSelections	No	Int	Number of lines or ways in the game (e.g. 100 for 100 lines, 5 for 243 ways)
availableBets	Yes	Array (int)	Array of integers of configured bets in ascending order. The bets are in credits
maxBet	Yes	Int	The maximum credit bet configured
availableDenoms	Yes	Array(float)	Array of configured decimal values for the value per credit in ascending order. The values are in cents
iconURLPrefix	Yes	String	The URL for retrieving the game’s icon for display on licensee site
availableFeatures	No	Array (string)	An array of strings indicating which bonuses are available within the game. Please see the table below regarding bonus types
newGame	Yes	Boolean	Indicates whether the game is a new game (< 1-2 months since release)
popular	Yes	Boolean	Indicated whether the game is popular (top 3-5 played games based on turnover)
availableRTP	Yes	Array (int)	A list of available RTP for a game

Game Types:

Type	Description
slots	Slots game
fish	Fishing game
table	Table game
vpoker	Video Poker game
pvp	Player VS Player game
other	Other games

Variant Types:

Type	Description
ways	Slots ways game (243, 1024, 4096, 720, etc…)
lines	Slots lines game
drop	For games that drop and win if contiguously attached symbols

Bonus Types:

Type	Description
fg	Free games bonus
bp	Bonus mini game
sk	Skill based mini game
br	Bonus reels (6th reel, bonus reels)
wl	Wheel based bonus games
spg	Standalone progressive
mpg	Linked mystery progressive
lpg	Linked non-mystery progressive

3.1 KA Games Icon URL:

The game icons can be retrieved and directly sourced from the KAGA system. E.g.

In the Game List Retrieval API, each game returns a game icon prefix. Additional parameters can be added to this URL to customize the look and feel of the launch icons.

There are four basic icon design types:

Type	Original Resolution	Example
rectangular	425 pixels x 265 pixels	
circular_framed	425 pixels x 425 pixels	
circular	425 pixels x 425 pixels	
square	425 pixels x 425 pixels	

oval	425 pixels x 224 pixels	
oval_framed	425 pixels x 223 pixels	

The game icon URL parameters available to pass

Parameter	Required	Description
game	Yes	The game ID. The game ID should match the values returned in the Game List Retrieval API. The value for "iconURLPrefix" returned using the Game List Retrieval API, this field is pre-populated with the correct game ID.
type	Yes	rectangular, circular_framed, circular, square – see examples in the table above
lang	No	Each icon is available in multiple languages. Please check with your sales contact or KA technical contact for the currently supported languages. Please pass the ISO-639-1 language code. If the language passed is not supported or if the lang parameter is not set, the default language of the image returned is English

If you require different image sizes, please scale the image dynamically on your site using standard HTML image size tags or Javascript. If image scaling is not an option, please get in touch with your KA sales or technical contact to arrange for a custom set of image icons.

4. KA GAMES LAUNCHER

All games must be launched with specifically formatted URL's. These URL's should be placed on the licensee page and linked to launching specific games from the KAGA library.

Game Launch URL Format:
https://<KAGASERVER>/?g=<GAME_NAME>&p=<PARTNER_NAME>&u=<UNIQUE_PLAYERID>&t=<UNIQUE_TOKEN>&ak=<PARTNER_ACCESSKEY>&loc=<PLAYER_LANGUAGE>&cr=<CURRENCY>&m=<GAME_MODE>&dn=<DENOMINATION>&o=<OPERATOR>&sn=<SOUND_OFF>&l=<LOBBY_URL>&sl=<SUPPORT_URL>&da=<DISPLAY_NAME>&ro=<REMOVE_OPTIONS>&asm=<AUTOSPIN_MODE>&if=<IFRAME_MODE>&mc=<BALANCE_METER_MODE>&v=<VERTICAL_MODE>&tl=<TITLE>&ld=<LOW_DEFINITION>&dc=<DISABLE_EXIT_CONFIRMATION&db=<DISABLE_INSUFFICIENT_BALANCE_ALERT>

Mandatory Parameters:

- **KAGASERVER:** The game launch URL for the partner. This value is retrieved from the game list command
- **GAME_NAME:** Unique name for the game being launched. The game names can be obtained from the "gamelist" REST service
- **PARTNER_NAME:** Unique name for the licensee (provided by KAGA during integration)
- **UNIQUE_PLAYERID:** Unique Identifier for the player. This can be the player's username, or any other persistent and consistent value used by the licensee to identify the player with RMP
- **PARTNER_ACCESSKEY:** Access key for the licensee (provided by KAGA during integration)
- **CURRENCY:** Currency to use for the player. Please use ISO 4217 currency codes
- **UNIQUE_TOKEN:** Unique token string generated by licensee that is unique for each Game Launcher link. The token will be sent back by RMP to the licensee during the "start" REST service and should be used when received by the licensee to verify that the player did launch the game

Optional Parameters:

- **GAME_MODE:** 0 for play for real, 1 for play for fun. Defaults to real game mode.
- **PLAYER_LANGUAGE:** Language to use for the player. Please use ISO 639-1 language codes (e.g. English is en). If not specified, KAGA RMP will auto detect the player's language. If the detected language is not supported, KAGA RMP will default to English. Note "zh" defaults to Simplified Chinese and Traditional Chinese is "zh_TW" or "zh-Hant"
- **OPERATOR_NAME:** Licensee identifier for sub-licensee operators
- **LOBBY_URL:** URL used for in game home button. If not specified, in game home button will either close the window if possible, or just end the player session. Setting this value to -1 will disable the in game Home button
- **DENOMINATION:** If specified, the launched game will be fixed denomination. If not specified, the player can change the denomination as needed directly inside the game
- **SUPPORT_URL:** The URL of a support web page to show the player if they want to file a support request. This can also be an email. The format of this field should be a qualified URL (e.g. https://partnersupporturl.com or mailto:support@partner.com)
- **SOUND_OFF:** Send 1 to turn sound off, or 0 to leave sound on. If not specified, sound defaults to ON
- **DISPLAY_NAME**: If a display name is specified, it will be shown inside the game screen.
- **REMOVE_OPTIONS**: This parameter can modify the drop down options that are available in the top right menu. 0 = all options available, 1 = Remove SUPPORT, EXIT, and FULLSCREEN, and GAME RESULT options. 2 = Remove SUPPORT, EXIT, and FULLSCREEN.
- **AUTOSPIN_MODE**: This parameter can modify the behavior of the AUTOSPIN button. 0 = allow for selection of # of autospins, 1 = disable AUTOSPIN, 2 = AUTOSPIN when enabled always results in UNTIL BONUS
- **LOW_DEFINITION**: This parameter when set to 1 will run the games at a lower resolution. This is useful for improving game performance on lower end computers and devices. By default this value is 0 which will run the games in high definition (retina) resolution
- **VERTICAL_MODE**: 0 = landscape mode, 1 = portrait mode. Default is landscape mode (0) if not specified.
- **TITLE:** This is the HTML title shown on tabs or windows. Default is KAGA88 if unspecified.

- **DENOM_HIDDEN:** 1 = hide denom controls and display of denom to player. Game should be set into fixed denom, if not, the default denom is used and cannot be changed. Default is visible (0) if not specified.
- **BALANCE_METER_MODE:** If this option is not set, or set to 0, then the game will start by showing the balance meter in cash denoted by the player's currency and the player can press the meter to toggle between showing credits and cash. When set to 1, the game will default to showing credits and the meter can be toggled between showing credits and cash. When set to 2, the game will default to showing credits and the meter cannot be toggled.
- **IFRAME_MODE**: If iframe is used please set this value to 1 to indicate special frame size handle for iframe under mobile device. And if **LOBBY_URL** is not being set when Home button is pressed a message "close_iframe" will post to parent window. Set the value to 0 if iframe is not applied.
  Note: To support full-screen function under Android and Desktop, please also add tags webkitallowfullscreen mozallowfullscreen oallowfullscreen msallowfullscreen allowfullscreen to the iframe tag.
- **DISABLE_EXIT_CONFIRMATION**: When set to 1, the exit confirmation will be displayed to the player when pressing the "Home" or "Exit Game" buttons
- **DISABLE_INSUFFICIENT_BALANCE_ALERT**: When set to 1, pop-up dialogs will not be displayed to player when trying to play a bet larger than balance. The game will not play.

Sample code of applying iframe:

```
<!DOCTYPE html>
<html>
<head>
   <meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>
<iframe id="gameframe" src="gameurl&if=1" webkitallowfullscreen mozallowfullscreen oallowfullscreen
msallowfullscreen allowfullscreen></iframe>

<script>
window.onmessage = function(e) {
   if (e.data == 'close_iframe') {
      var gameframe = document.getElementById('gameframe');
      gameframe.src = 'about:blank';
   }
};

function updateiFrameSize() {
   var iFrameID = document.getElementById('gameframe');
   if(iFrameID) {
      iFrameID.style.width = window.innerWidth + "px";
      iFrameID.style.height = window.innerHeight + "px";
      iFrameID.contentWindow.postMessage(window.innerWidth + "." + window.innerHeight, '*');
   }
}

updateiFrameSize();
window.onresize = function() {
   updateiFrameSize();
   clearTimeout(timeUpdate);
   timeUpdate = setTimeout(function() {
      updateiFrameSize();
   }, 200); // for iPhone 5 will have a slower update of width / height
};
</script>
</body>
</html>
```

The Game Launch URL can be opened by the player in the following formats:

- PC & Mac
    - New browser tab
    - New browser window
        - Please ensure a 3:2 or 16:9 aspect ratio window. 1136x640 (16:9) and 960x640 (3:2) without browser bar, search bars, location bars, and browser tool bars are the preferred resolution and styles for the new window. The games look best in a 16:9 aspect ratio.
    - frame or iframe in existing licensee web page
    - Load directly over the licensee web page
        - It is possible to have the player sent back to a licensee specified URL after the player closes or exits the game by specifying the LOBBY_URL parameter in the Game Launch URL
- Mobile browsers
    - New browser tab
    - Load directly over the licensee web page
        - It is possible to have the player sent back to a licensee specified URL after the player closes or exits the game by specifying the LOBBY_URL parameter in the Game Launch URL

All game launch URL's must utilize SSL and the URL service prefix should be "https". This ensures end to end security for the game.

It is recommended that the games launch in a new browser window or tab using Javascript window.open() commands. This allows the game to be exited gracefully when the player decides to either close the new window or tab. For scenarios where the game cannot be open in a new window or tab, passing the LOBBY_URL in the Game Launch URL is recommended.

Each game launch starts a new session for the player. It is possible for a single player to have multiple sessions open simultaneously while playing different KAGA provided games. Each open game and game variation that is being played will consist of a session and will need to be launched using different launch URL's linked from the licensee's site. Please note that each game can only be launched uniquely per player and currency type combination.

The denomination parameter used to set a single, fixed, game denomination is ignored if the launched game is resuming a previously uncompleted bonus. In this scenario, the denomination played during the bonus triggering wager is used as the single fixed denomination for the session.

KA Gaming

5. SEAMLESS WALLET REST INTEGRATION

KA offers a seamless wallet integration that allows the synchronization of the player balance across all games and systems utilized by the licensee and any of its operators. The seamless wallet integration is recommended as it offers an effortless game play experience for the player. However, KA does provide a Deposit/Withdraw wallet model (non-seamless) mechanism as well. Please see the section titled Detached Wallet Integration for details about how to integrate KA's non-seamless wallet.

A set of REST web services needs to be built by the licensee according to KAGA specifications that allow the synchronization of bets, balance, and winnings for players while playing integrated games. These services are called while the player is inside and playing a game as part of the overall game transaction. The web services built by the licensee will need to be able to perform with low latency and high concurrency in order to ensure a smooth playing experience for the player.

5.1 Seamless Wallet Integration Overview

KAGA RMP and licensee wallet integration enables the seamless play of KAGA games by players of the licensee's site(s). The integration covers the updating of balance, stakes, and game sessions of a player playing KAGA games as initiated by the player on the licensee's site.

A series of REST web services are provided as part of this integration. These services are:

Service	Purpose	General Usage
start	Setup and start a new KAGA game session	RMP request to licensee at start of a game
balance	Get the player's current balance	RMP requests to licensee to sync up game RMP game balance with actual balance of player
play	Transmit details of a game play/wager	RMP requests to licensee before every wager to verify sufficient player balance and permission to play.
credit	Add winnings to player balance	RMP requests to licensee after any win to credit the player with winnings
end	The end of a player's game session	RMP requests to licensee if a player is idle timed out or manually opts to exit the game
revoke	Cancel a previous bet or credit service call	RMP requests to licensee to cancel a previous transaction

gameList	Get list of all games and titles available to licensee	Licensee calls RMP to get an updated list of all configured and available games
playerReport	Get player game spin and bonus pick report	Licensee calls RMP to get an updated list of game spin and bonus pick record report
summaryReport	Get plaer game summary report	Licensee calls RMP to get player summary report
gamePlayerReport	Get specific game spin and bonus pick report	Licensee calls RMP to get an updated list of game spin and bonus pick record report
playerGamePlayReport	Get specific player game spin and bonus pick report	Licensee calls RMP to get an updated list of game spin and bonus pick record report

5.2 Flow Sequence Example:

1. Licensee calls RMP to get latest available game list and configuration details
2. Licensee generates page showing links to KAGA web games. Each link is uniquely generated and formatted to identify a specific game, a player, the licensee, currency used, language, and a unique token.
3. Player clicks on a link to a specific game generated on the licensee page
4. Player launches the game in their browser.
5. When launched, RMP contacts Licensee via the “start” REST service to transmit startup details, player validity, and player balance.
6. Player plays a spin or places a wager
7. When a wager is placed, RMP contacts licensee via the “play” REST service to validate the user has enough balance and permissions to play a spin.
8. Player wins a bonus mini game (not of free spins variety) and starts playing bonus game
9. Each bonus mini game win results in the “credit” REST service being called by the RMP to the licensee to notify of balance changes based on the player’s bonus mini game wins.
10. Player takes a break but does not close the game. After going idle for > 1 minute, RMP will send a “balance” REST service request to licensee to ensure RMP game balance is synchronized with licensee player balance. After 15 minutes of idle, the player is automatically logged out.
11. Player returns to the game within 15 minutes and closes the game browser, tab, window, or presses the exit button inside the game.
12. RMP sends an “end” REST request to the licensee to indicate that the player has ended his game session

5.3 Request Format

All messages are sent directly via HTTPS requests and in JSON encoded format in line with standard REST web service practices and frameworks.

All messages utilize **Content-Type:** application/json

All Wallet API messages contain:

- "hash" request parameter sent using GET on the Request URL. This is used for message security and validation. See the section titled Wallet API Security to learn more how to utilize the "hash" parameter
- JSON formatted object send directly via POST

5.4 Seamless Wallet API Security

KAGA utilizes a layered approach to security. These focus on authenticating RMP wallet requests with the licensee, high strength transmission encryption, and digital signatures.

The implementation of layered security is:

- Licensee exposed Wallet REST services must be over HTTPS only with a valid SSL certificate
    - Self-signed certificates can be used for test environments
- Licensee should whitelist only KAGA server IP's and prevent calls to the REST API's from any unrecognized IP sources
- Every wallet request sends the player's IP address (IPV4 or IPV6). The licensee should verify that the IP address JSON field sent by KAGA RMP matches the actual player IP address
- Every request needs to be validated using an HMAC hash encoded with SHA256 64 character long hexadecimal string sent directly on the request URL.
    - The hash is created using the secret key shared between the licensee and RMP and the entire contents of the REST JSON message string including any enclosing braces in the JSON message.
        - PHP example: hash_hmac('SHA256', $message, $key)
        - Python example: hmac.new('secretKey', msg='{ test: "json", use: "actual" }', digestmode=hashlib.sha256).hexdigest()
        - Additional examples with fully operational code snippets can be provided on request
    - The hash is sent directly in the URL as a query parameter outside of the API's JSON message payloads. Example:
        - https://licensee_url/start?hash=5132ai491887bcdff62ea239deacc34719305749201267929ea476c3928176cd

There is common web service JSON parameters that are transmitted from RMP to the licensee as part of every RMP initiated wallet service message. Similarly, there are also a set of common

Wallet Response parameters that should be included in every response message sent back by the licensee.

Currency Formatting:
All Wallet API balances and values should be returned as ***CENTS.*** For usage of currencies that do not utilize fractions, please multiply by 100. E.g. JPY 1000 should be returned as 100000.

5.5 Balance Sequencing:

An optional balanceSequence parameter can be sent back in all responses where balance is also sent. If used, this sequence number is used to ensure that the latest balance response is used. In the scenario where multiple balances are returned for a player simultaneously, the one with the highest sequence number is updated to the game clients.

Common Seamless Wallet Request JSON Parameters:

Parameter	Required	Type	Usage
timestamp	Yes	Long	Timestamp for the request in milliseconds Epoch time (UTC milliseconds since 1/1/1970)
sessionId	Yes	String (32 chars)	Identifies the player session as defined by each game launched. This parameter is generated by RMP during game launch and
partnerPlayerId	Yes	String	The UNIQUE_PLAYERID field specified in the Game Launch URL. This field should be a unique identifier used by the licensee to identify a player and persist beyond a single game session
currency	Yes	String (ISO 4217)	The CURRENCY field specified in the Game Launch URL. This is the currency used by the player.
gameId	Yes	String	The GAME_NAME field specified in the Game Launch URL. This is the unique identifier for a licensed game made available to or enabled by the licensee
token	No	String	The same token that was sent in the Game Launch URL and generated by the licensee if available
operatorName	No	String	The OPERATOR field specified in the Game Launch URL if available. This field is used by the licensee to distinguish between licensee's own sub-brands/sub-licensees.
action	Yes	String	Either start, balance, play, credit, end, revoke to identify the message type.
playerIp	Yes	String	Player IP. Note the IP field may be IPV4 or in IPV6 format

Common Seamless Wallet Response JSON Parameters:

Parameter	Required	Type	Usage
status	Yes	String	A status string or message. If the message was a success, please send back "success" as the status. For other errors, send back a specific error message in the status field. This error message won't be shown to the player but tracked by RMP to aid in any issue resolutions
statusCode	Yes	Int	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
userMessage	No	String	If specified, this message gets shown to the user in a popup dialog within the game. This can be used to inform the user about their account status, an error, or of bonuses given them real-time

5.6 Seamless Wallet Status Codes:

Status Code	Action	Usage
0	All	Request successfully processed
1	All	Exception or error in request handling. Please specify a reason or error message in the "status" parameter
2	All	Invalid request – missing parameters in the request or invalid format
3	All	Invalid hash. Send if security hash validation fails
4	All	Invalid player. Send if player IP address is mismatched or player Id not found
5	All	Request mismatch. Used if player details sent by RMP do not match data for player on licensee
100	start	Invalid token
101	start	Token already used
200	play	Insufficient balance (cashable or bonus) to play bet
201	play	Wager permission denied or blocked by Licensee
300	credit	Transaction does not exist
301	credit	Licensee or operator denied crediting to player (cashable or bonus) balance
400	revoke	Transaction does not exist
401	revoke	Transaction no longer revocable
500	end	Session does not exist
600	gameList, playerReport	Invalid partner

2002	playerReport	Invalid partner player

Request/Response Examples

Request URL:

```
https://licensee_server/play?hash=663187fac4edbcdff62ea239deacc34719305749201267929ea476c3928176cd
```

Request POST Content JSON:

```
{
"transactionId":"672f27df1dcc401fa4ed6cdc14fb78f8",
"betAmount":200,
"winAmount":25,
"jpc":0,
"selections":5,"
betPerSelection":5,
"freeGames":false,
"sessionId":"375e3e418a45494c92bf1e6ec2f7460e"
,"partnerPlayerId":"demouserf",
"action":"play",
"gameId":"Vampire",
“timestamp”:  1503383341514
}
```

Response Content JSON:

```
{
"balance":7615,
"status":"success",
“statusCode”: 0
}
```

5.7 Start Request

This is sent to start a new game session. The wallet system of the licensee should do the following steps:

1. Receive KAGA JSON request
2. Verify HMAC of JSON body
3. Retrieve and verify details of unique token matches the same details
    a. PlayerIp
    b. Details stored for the Token match playerId, playerIp, game launched
    c. Verify player has permission to play

4. Once verification is complete, responding as “success” allows the player to successfully enter the game from the Game Launch URL
5. Send an appropriate error status code should any of the processing to start a game session fails or is invalid

KAGA sends these additional parameters (common ones detailed in section above are also sent)

Parameter	Required	Type	Usage

Successful Licensee Response (in addition to common response parameters):

Parameter	Required	Type	Usage
playerId	Yes	String	The Player ID sent to the Game Launch URL
sessionId	Yes	String	Session ID sent as part of the common parameters in the start request
balance	Yes	Long	Player’s balance in cents (e.g. $10,000.00 = 1000000)
balanceSequence	No	Long	An incrementing sequence # for balance responses or sends to RMP
currency	Yes	String	ISO 4217 currency code
sessionRTP	No	Int	This is the desired RTP applicable to only this game session. Note that it is possible for the same player to have multiple game sessions open to different games playing at different RTP’s. If this value is not specified, the RTP used for the player’s game session will utilize the defaults that are set for the player, operator, game, or system wide. To get a list of available RTP’s per game, please see the gameList API.

5.8 Play Request

The play request is sent whenever a player attempts to play a wager or spin. Each non-free game play request will start a new and unique transaction Id. In cases of free games, a specific parameter will be passed to the licensee to indicate that the game is part of a free game and the licensee should not deduct from the player’s balance for this spin. The transaction Id sent in free game play requests match the transaction ID in the triggering wagered play

The licensee server should handle the play request as follows:

1. Receive KAGA JSON play request
2. Verify HMAC of JSON body
3. Verify player details
4. Verify player has sufficient balance to play
5. Verify player has permission to play

6. Determine if player balance deduction is from bonus balance or cashable balance
7. Deduct bet amount from balance
8. Add win amount back to player's balance
9. Respond to KAGA RMP with player's new balance
10. If any error occurs, respond with a corresponding error status code

KAGA sends these additional parameters (common ones detailed in section above are also sent)

Parameter	Required	Type	Usage
transactionId	Yes	String	Transaction identifier for the current wager
betAmount	Yes	Long	Amount player is attempting to wager in cents
winAmount	Yes	Long	Amount player won in this wager in cents
selections	Yes	Int	Number of lines or ways wagered
betPerSelection	Yes	long	Bet multiplier wagered for each line or ways in cents
freeGames	Yes	Bool	If play event was a free game event. If it is a free game event, licensee should not deduct bet amount and only add the win amount back to the player's balance
round	Yes	Int	The step within a transaction. E.g. if player wins 12 free games and the play request is for the 3rd free game, the round will be set to 3. If there are no additional rounds to a given wager or it is the first event in a wager, then the round will be 0
roundsRemaining	Yes	Int	The number of un-played rounds remaining. This value is 0 if it is the last round of a free game or if there are no free game is triggered.

Successful Licensee Response (in addition to common response parameters):

Parameter	Required	Type	Usage
balance	Yes	Long	The player's new balance in cents. Please return this value even if the player does not have enough balance to play the spin so the KA game client will use this value to re-set the actual balance displayed to the player.
balanceSequence	No	Long	An incrementing sequence # for balance responses or sends to RMP
jackpotWon	No	Long	Value of jackpot won in cents
jackpotWonClearBaseGame	No	Boolean	Clear base game results and reward jackpot only
partnerBonusPlay	No	Int	To indicate the wager is a rewarded game (free play), and two types are supported: 1. The rewarded belongs to cash back 2. The rewarded belongs to bonus

partnerBonusBalance	No	Long	The player's current rewarded balance remaining, in cents

5.9 Credit Request

The credit request is sent whenever a player wins an amount outside the course of a normal spin or free spin. Most frequently, it is called when a player has entered any bonus mini games or bonus pick events. The transactionId of credit requests will match the transactionId of the triggering spin/wager that caused subsequent balance crediting events.

The licensee server should handle credit requests as follows:

1. Receive KAGA JSON credit request
2. Verify HMAC of JSON body
3. Verify player details
4. Verify player still has permission to play
5. Add credit request amount to player balance
6. Respond to KAGA RMP with player's new balance
7. If any error occurs respond with a corresponding error status code

KAGA sends these additional parameters (common ones detailed in section above are also sent)

Parameter	Required	Type	Usage
transactionId	Yes	String	Transaction identifier
amount	Yes	Long	Amount to add to player balance in cents
type	Yes	String	The type of event that triggered the credit request. E.g. "BonusPick" or "Progressive"
creditIndex	No	Int	The index of a bonus pick or the level of a progressive that caused this credit request.

Successful Licensee Response (in addition to common response parameters):

Parameter	Required	Type	Usage
balance	Yes	long	The player's new balance in cents
balanceSequence	No	Long	An incrementing sequence # for balance responses or sends to RMP

5.10 Revoke Request

A revoke request is sent if during the course of a transaction, the player disconnects and the transaction should be canceled.

The licensee server should handle revoke requests as follows:

1. Receive KAGA JSON revoke request

2. Verify HMAC of JSON body
3. Verify player details
4. Revoke the specified transaction round and all rounds with a value greater than the revoked round

KAGA sends these additional parameters (common ones detailed in section above are also sent)

Parameter	Required	Type	Usage
transactionId	Yes	String	Transaction identifier
round	Yes	Int	The round to cancel

Successful Licensee Response (in addition to common response parameters):

Parameter	Required	Type	Usage
balance	Yes	long	The player's new balance in cents
balanceSequence	No	Long	An incrementing sequence # for balance responses or sends to RMP

5.11 Balance Request

A balance request is sent by KAGA RMP to synchronize and get the latest player balance from the licensee. This could be sent at any time during the player's session.

The licensee server should handle balance requests as follows:

1. Receive KAGA JSON balance request
2. Verify HMAC of JSON body
3. Verify player details
4. Respond with the most current player balance

KAGA sends no additional parameters aside from the common request parameters.

Successful Licensee Response (in addition to common response parameters):

Parameter	Required	Type	Usage
balance	Yes	long	The player's new balance in cents
balanceSequence	No	Long	An incrementing sequence # for balance responses or sends to RMP

5.12 End Request

The end request is sent when the player exits the game. It will send only the fields that are common to all seamless wallet requests. Please respond with a statusCode of 0 and status of "success" to acknowledge this message.

6. DETACHED WALLET INTEGRATION

KA offers a detached wallet solution where a set amount of cash specific to a currency, licensee, operator and game are deposited for a player. The player can choose to withdraw the funds or the amount is automatically withdrawn when the player exits the game.

There are two modes of operation for detached wallets. One is to have a wallet for each combination of game, player, and currency where two different KA games played by the same player will utilize separate detached wallets. The second mode allows all KA games to share one wallet for each combination of player and currency.

A series of REST web services are hosted by KA as part of this integration. These services are:

Service	Purpose	General Usage
deposit	Setup and start a new KAGA game session	Licensee calls this service on RMP to deposit funds into a player's game wallet. If the player is starting the game for the first time, a new wallet is also created as part of the deposit process. Note this API will also return the online status of a player indicating whether he/she is online and currently playing a KA game.
balance	Get the player's current balance	Licensee calls this service on RMP to get the current wallet balance for a player's game wallet. Note this API will also return the online status of a player indicating whether he/she is online and currently playing a KA game.
withdraw	Transmit details of a game play/wager	Licensee calls this service on RMP to withdraw an amount from the player's game wallet.
history	Retrieve all events associated with a specific wallet	Licensee calls this service on RMP to return all events that occurred for a specific wallet. This includes a full accounting of all deposits, withdrawals, and game plays that interact with the player balance
checkTransaction	Check the status of a previous wallet transaction	Licensee can call this API to check the status of a previous deposit or withdrawal of player balance. This also allows licensee to check the

		transaction status of any previous game plays.
close	Automatically deposit back wallet balance to the licensee	RMP sends to the licensee the entirety of the player's balance and closes out the wallet. This is called when the player exits the game or the game logs the player out automatically when the player goes idle

6.1 General Detached Wallet Usage Flow (Separate Wallet for Each Game)

1. Licensee calls RMP to get latest available game list and configuration details
2. Licensee generates page showing links to KAGA web games. Each link is uniquely generated and formatted to identify a specific game, a player, the licensee, currency used, language, and a unique token.
3. Player clicks on a link to a specific game generated on the licensee page and is taken to the licensee's deposit UI
4. Player decides how much to deposit into the game
5. Licensee site calls the "deposit" REST service on RMP and deposits/transfers over the desired amount
6. Licensee site launches the game in the Player's browser using the generated game launcher URL (see the section about generating game launcher URL's)
7. Game launches and the player is shown the deposited balance inside the game.
8. Player plays a spin or places a wager
9. When a wager is placed, RMP deducts the amount from the detached wallet.
10. When a spin or bonus game is won, RMP credits the amount into the detached wallet
11. Player takes a break but does not close the game. After 15 minutes of idle, the player is automatically logged out.
12. If the licensee has set a "close" callback, RMP sends "close" REST request to the licensee to indicate that the player has ended his game session, withdraws the entire wallet balance and pushes the balance back to the licensee. If the licensee has not configured a "close" callback, then no player balances will be transferred back to licensee.

6.2 General Detached Wallet Usage Flow (Shared Wallet for all KA Games)

1. Player goes to licensee site and does a fund transfer (deposit) to play KA games. Licensee site calls the "deposit" REST service on RMP and deposits/transfers over the desired amount.
2. Licensee calls RMP to get latest available game list and configuration details

3. Licensee generates page showing links to KAGA web games. Each link is uniquely generated and formatted to identify a specific game, a player, the licensee, currency used, language, and a unique token.
4. Player clicks on a link to enter a specific game URL generated on the licensee page
5. Licensee site launches the game in the Player's browser using the generated game launcher URL (see the section about generating game launcher URL's)
6. The game launches and the player is shown the deposited balance inside the game.
7. Player plays a spin or places a wager
8. When a wager is placed, RMP deducts the amount from the detached wallet.
9. When a spin or bonus game is won, RMP credits the amount into the detached wallet

6.3 Detached Wallet API Security

KAGA utilizes a layered approach to security. These focus on authenticating RMP wallet requests with the licensee, high strength transmission encryption, and digital signatures.

The implementation of layered security is:

- KA exposed Wallet REST services are available only over HTTPS with valid SSL certificates
- KAGA will whitelist all licensee IP's for access to KAGA hosted detached wallet REST services
- Every request sent to KAGA from the licensee as well as sent to licensee KAGA needs to be validated using an HMAC hash encoded with SHA256 64 character long hexadecimal string sent directly on the request URL.
    - The hash is created using the secret key shared between the licensee and RMP and the entire contents of the REST JSON message string including any enclosing braces in the JSON message.
        - PHP example: hash_hmac('SHA256', $message, $key)
        - Python example: hmac.new('secretKey', msg='{ test: "json", use: "actual" }', digestmode=hashlib.sha256).hexdigest()
        - Additional examples with fully operational code snippets can be provided on request
    - The hash is sent directly in the URL as a query parameter outside of the API's JSON message payloads. Example:
        - https://rmp_url/kaga/wallet/deposit?hash=5132ai491887bcdff62ea239deacc34719305749201267929ea476c3928176cd

Currency Formatting:

All Wallet API balances and values are transmitted and represented in ***CENTS.*** For usage of currencies that do not utilize fractions, please multiply by 100. E.g. JPY 1000 should be returned as 100000.

6.4 Detached Wallet Status Codes:

Status Code	Action	Usage
0	All	Request successfully processed
1	All	Exception or error in request handling. See reason specified in the "status" parameter
2	All	Invalid request – missing parameters in the request or invalid format
3	All	Invalid hash. Security hash validation failed
4	All	Wallet not found specific to a licensee, operator, and game
5	All	Request mismatch. Used if player details sent by RMP do not match data for player on licensee
10	withdraw	Withdrawal request failed because of insufficient funds in player's game wallet

6.5 Common Detached Wallet Request JSON Parameters:

There are common web service JSON parameters that are transmitted from RMP to the licensee as part of every RMP initiated wallet service message. Similarly, there are also a set of common Wallet Response parameters that should be included in every response message sent back by the licensee.

Parameter	Required	Type	Usage
username	Yes	String	A unique player identifier used by the licensee
partnerName	Yes	String	This is the name of the licensee defined by RMP. The value used for this field should be the same as provided to you by the KA team.
gameName	No	String	This field is the name for the game used by the wallet. This game name should be the same as the game names used in the Game Launch URL and retrieved from the Game List API. If no game name is specified, then the wallet is considered shared by all KA games as described in the sections above.
currency	Yes	String	ISO 4217 currency code. If currency is not specified, it will return data related with all currency

accessKey	Yes	String	The access key for the licensee (provided by your KA rep)
operatorName	No	String	The operator, agent, sub-agent, or brand used to identify players.

6.6 Common Detached Wallet Response JSON Parameters:

Parameter	Required	Type	Usage
status	Yes	String	A status string or message. If the message was a success, please send back "success" as the status. For other errors, send back a specific error message in the status field. This error message won't be shown to the player but tracked by RMP to aid in any issue resolutions
statusCode	Yes	Int	A shared numerical error code. Please refer to the section on Detached Wallet status codes to see which are available and their usage
balance	Yes	Long	The current wallet balance
walletTransactionId	No	String	The wallet transaction id for the action taken on the wallet. Deposit and Withdrawal wallet events will have a walletTransactionId. History and Balance requests will not.

6.7 Deposit Request

URL Format:
https://RMP_URL/kaga/wallet/deposit?hash=<HMAC>

This REST request is sent from the licensee to RMP to deposit an amount specific to a player, currency and game. This should be called BEFORE the player launches the game in their browser. The wallet system of the licensee should do the following steps:

1. Player clicks a game link on licensee site
2. Licensee presents player with UI to input or determine how much of their balance should be deposited into the game selected to launch
3. Licensee calls "deposit" REST service to KA RMP and passes along the amount player selected to deposit. The REST request should be sent with a GET parameter named "hash" that is generated as described above
4. KA will verify HMAC of JSON body

5. KA will retrieve an existing wallet specific to that player and game. If one does not exist that is specific to the player and game, a new wallet is created
6. KA will add the deposit amount to the player's wallet
7. KA will respond to the "deposit" rest request with a status code
8. If the status is successful, the licensee should proceed with launching the game for the player using a generated Game Launcher URL (see section above regarding generating Game Launcher URL's)

This API will also return the online status of a player indicating whether he/she is online and currently playing a KA game.

Send to KAGA RMP these additional parameters including all of the common ones detailed in section above

Parameter	Required	Type	Usage
depositAmount	Yes	Long	This is the amount to deposit for the player's game wallet
externalTransactionId	No	String	This is the licensee's transaction id. This field is optional.

In addition to the common Detached Wallet Response parameters, KAGA also sends the following parameters in response to the Deposit Request:

Parameter	Required	Type	Usage
online	Yes	Boolean	True/false if the player is currently online and playing a KA game
newWallet	Yes	Boolean	True/false if the player is just created currently
externalTransactionId	No	String	This is the licensee's transaction id received from the request parameter.

6.8 Balance Request

URL Format:
https://RMP_URL/kaga/wallet/balance?hash=<HMAC>

This REST request is sent from the licensee to RMP to retrieve the balance for a player's game wallet. This can be sent when the player session is active, inactive, or at any time to check the balance of a player's game wallet.

This API will also return the online status of a player indicating whether he/she is online and currently playing a KA game.

The Balance REST request only utilizes the Common Detached Wallet parameters for the request.

In addition to the common Detached Wallet Response parameters, KAGA also sends the following parameters in response to the Balance Request:

Parameter	Required	Type	Usage
online	Yes	Boolean	True/false if the player is currently online and playing a KA game

6.9 Withdraw Request

URL Format:
https://RMP_URL/kaga/wallet/withdraw?hash=<HMAC>

This REST request is sent from the licensee to RMP to withdraw a specified amount for a player's game wallet. This can be sent when the player session is active, inactive, or at any time to check the balance of a player's game wallet.

This API will also return the online status of a player indicating whether he/she is online and currently playing a KA game.

Send to KAGA RMP these additional parameters including all of the common ones detailed in section above

Parameter	Required	Type	Usage
withdrawalAmount	Yes	Long	This is the amount to withdraw from the player's game wallet
externalTransactionId	No	String	This is the licensee's transaction id. This field is optional.

In addition to the common Detached Wallet Response parameters, KAGA also sends the following parameters in response to the Withdraw Request:

Parameter	Required	Type	Usage
online	Yes	Boolean	True/false if the player is currently online and playing a KA game
externalTransactionId	No	String	This is the licensee's transaction id received from the request parameter.

6.10 CheckTransaction REQUEST

URL Format:
https://RMP_URL/kaga/wallet/checkTransaction?hash=<HMAC>

This REST request is sent from the licensee to RMP to check the status of a previously run transaction for a specific wallet. If the transaction being queried failed, checkTransition will return an error indicating that the transaction is not found.

Send to KAGA RMP these additional parameters including all of the common ones detailed in section above

Parameter	Required	Type	Usage
walletTransactionId	Yes	String	This is the transaction id for a previously run wallet transaction. This field supports both KAGA RMP wallet transaction id and licensee's transaction id.

KA RMP responds to checkTransaction requests with all of the common parameters detailed in the section above in addition to the following parameters:

Parameter	Required	Type	Usage
date	Yes	Date	This is the date that the transaction took place
ip	Yes	String	This is the IP address from where the transaction was initiated
event	Yes	String	This is the event type of this transaction. Valid values are Deposit, Withdrawal, Start, GamePlay, FreeGamePlay, Credit
balanceAtTimeOfTransaction	Yes	Long	The wallet balance at the time of the transaction (in cents)
debit	Yes	Long	The amount debited (in cents) for a wallet event. Note that this amount is in the same currency as the one specified in the History request and the game/player/currency specific wallet
credit	Yes	Long	The amount credited (in cents) for a wallet event. Note that this amount is in the same currency as the one specified in the History request and the game/player/currency specific wallet
externalTransactionId	No	String	This is the licensee's transaction id.

6.11 History Request

URL Format:
https://RMP_URL/kaga/wallet/history?hash=<HMAC>

This REST request is sent from the licensee to RMP to get the transaction history for a wallet. Transaction events include all deposits, withdrawals, game play debits/credits, and all game start events utilizing the game/player specific wallet.

Send to KAGA RMP these additional parameters including all of the common ones detailed in section above

Parameter	Required	Type	Usage
startDate	No	Date	A starting date to limit wallet events in the response
endDate	No	Date	An ending date to limit wallet events in the response

KA RMP responds to history requests with all of the common parameters detailed in the section above in addition to the following parameters:

Parameter	Required	Type	Usage
Events	Yes	WalletEvent	Returns all wallet events that have occurred for a specific wallet within a specific date (if specified)

WalletEvent Type:

Parameter	Required	Type	Usage
date	Yes	Date	The date of the wallet event
ip	Yes	String	The IP address that triggered the wallet event. This IP address could be IPv4 or IPv6 depending on the type of network utilized that triggered the wallet event
walletTxId	Yes	String	The wallet transaction ID for the wallet event that occurred
event	Yes	String	The event type. Normal event types are: Deposit, Withdrawal, Start, GamePlay, FreeGamePlay, Credit
balance	Yes	Long	The wallet balance at the time of the event (in cents)
Debit	Yes	Long	The amount debited (in cents) for a wallet event. Note that this amount is in the same currency as the one specified in the History request and the game/player/currency specific wallet
Credit	Yes	Long	The amount credited (in cents) for a wallet event. Note that this amount is in the same currency as the one specified in the History request and the game/player/currency specific wallet

gameTransactionId	No	String	A transaction ID for a game play event. This is null for wallet interactions that are not triggered from a game play event
game	Yes	String	Game name

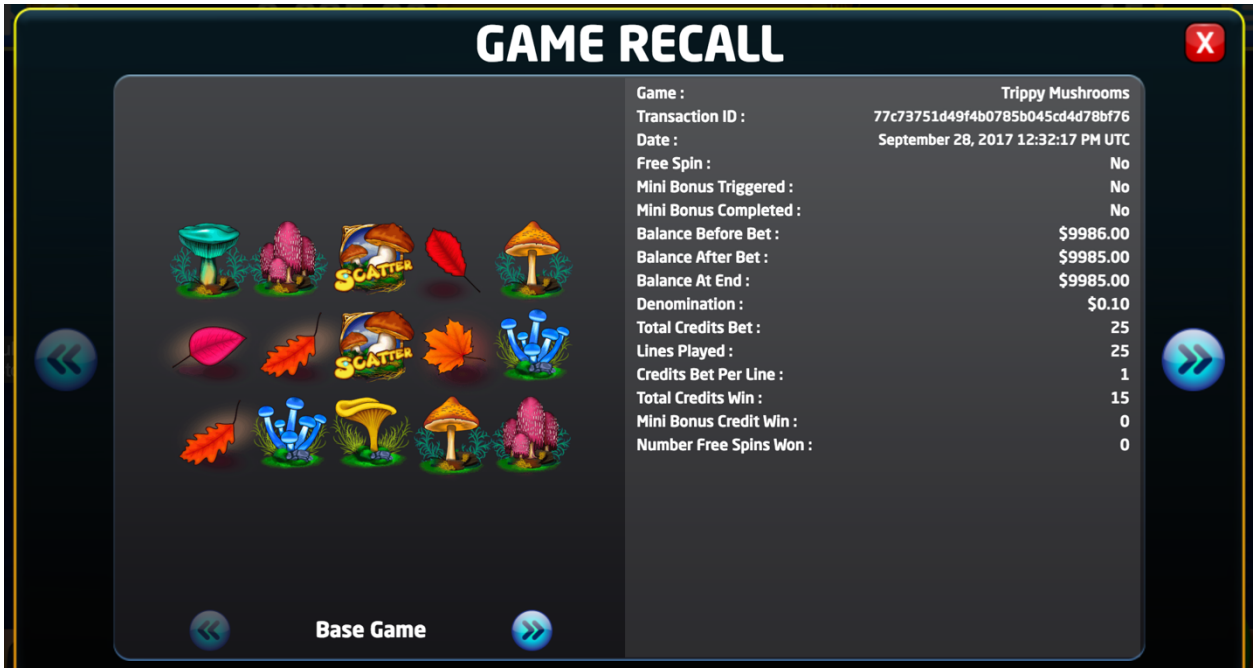
7. REPORTING INTEGRATION

A set of REST API's are available on the RMP server for call by the licensee to retrieve player game history details and overall licensee play details within a specific time range. The licensee may use these API's to perform game recall on actual player wagers as well as find their winnings and losses for specific time ranges.

In addition to the set of server to server REST API's available to look up player game history, a graphical game recall is also available for embedding into an iframe or opened as a separate link that displays the game outcomes for a given game transaction id.

7.1 Game Recall Graphic Link

The graphical game recall is a URL that can be embedded into an iframe or opened as a separate tab/window to show the game outcome for a given game transaction id.



The graphical game recall link needs to be constructed by the partner in the following format: https://kaga_gameURL/?g=<GAME_NAME>&p=<PARTNER_NAME>&ak=<ACCESS_KEY>&loc=<LANGUAGE>&grid=<GAME_TRANSACTION_ID>&grts=<SECONDS_SINCE_EPOCH_UTC>grha=<HMAC>

Normally the URL domain name (kaga_gameURL in the format example above) is the same URL domain/prefix the licensee uses for the beginning part of the gameLaunchURL.

HTTP GET Parameter	Description	Required
g	This is the programmatic name for the game. The same name used in the gameLaunchURL should be used here	Yes
p	This is the partner name. The same partner name used to launch the games in the gameLaunchURL should be used for this value as well	Yes
ak	This is the licensee's access key. Again, this value should mirror the accessKey used in the gameLaunchURL for launching games for the client	Yes
loc	This is the language that should be used for displaying the contents in the graphical game recall. This should follow ISO-639-1. For Traditional Chinese, please use "zh-Hant" or "zh_TW" and for Simplified Chinese use "zh-Hans" or "zh_CN"	No
grid	This is the transaction Id for the spin to query for. The transaction ID can be obtained either through the responses during a game play for seamless wallet integrations, or querying the wallet event for detached wallets	Yes
grts	This is the current UTC seconds since the epoch (1970/1/1). This field is used to determine the validity of a link. Each link is only valid for 15 minutes from the time of generation to the time of it being used.	Yes
grha	This is an HMAC field used to secure graphical game recalls. Please create an HMAC for the following fields concatenated together GAME_TRANSACTION_ID + SECONDS_SINCE_EPOCH and using the shared secret key.	Yes

HMAC examples for Graphical Game Recall:

- PHP example: hash_hmac('SHA256', $transactionId + $epochSeconds, $secretKey);
- Note the sequence of the fields concatenated as part of the HMAC message is important. Please ensure that the sequence is TRANSACTION_ID concatenated with no spaces or separating characters with the number of seconds since the epoch.

7.2 Game Recall Rest Server To Server Request

Game Recall REST services can be used by the licensee to retrieve the results of a specific game transaction and all rounds and bonuses resulting from the game transaction.

The direction of this call is for the licensee to call into the gameRecall REST API hosted on the KAGA RMP server.

Content-type: application/json
URL: https://kagarmp_server/kaga/gameRecall?hash=<HMAC>

Game Recall POST JSON Parameters:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
transactionId	Yes	String	The transaction ID for a game play/spin
tzOffset	No	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, please use -4, UTC+8 please use 8, UTC+0, please use 0) This is used to setup spinDate, pickDate with timezone. The default value is 0

Game Recall JSON Response Parameters:

Parameter	Required	Type	Usage
es	Yes	String	A status string or message.
ec	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
totalCashWon	Yes	Long	The total cash value won in cents
totalCashBet	Yes	Long	The total cash value wagered in cents
currency	Yes	String	The currency used for this transaction (ISO 4217)
gameDetails	Yes	Array	Array of SpinReport (see SpinReport parameters detailed below). This can contain all free game rounds resulting from a wagered transaction. If only a base game is played, then there is only 1 record in the gameDetails array. If there are subsequent free spins, all played free spin rounds are also included in the gameDetails Array
bonusPicks	No	Array	Array of BonusPickReport (see BonusPickReport parameters detailed below).

SpinReport Parameter Structure

Parameter	Required	Type	Usage
playerId	**Yes**	String	Player ID shall be unique to represent a player
operator	No	String	Operator or agent name for player if any
transactionId	Yes	String	The transaction Id to identify a spin. Free spin will share the same transaction Id with the dependent normal spin to identify as a "Coin Out"
gameName	Yes	String	The name of the game played
gameType	Yes	String	The type of the game. Valid game types are (“slots”, “fish”, “vpoker”, “pvp”, “other”, “table”)
spinDate	Yes	String	The date time of the play. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-21 15:56:18)

tzOffset	Yes	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, use -4, UTC+8 use 8, UTC+0, use 0)
currency	Yes	String	ISO 4217 currency code
rtp	Yes	Float	Return to Player
denomination	Yes	Float	The unit of money to played or won
selections	Yes	Integer	Number of lines or ways
creditPerSelection	Yes	Long	Credit bet per selection
cashPlayed	Yes	Long	Cash played for single spin (in cents)
cashPlayedRaw	Yes	Long	Cash played for single spin (in cents) without partner currency multiplier
cashWon	Yes	Long	Cash won for single spin (in cents)
cashWonRaw	Yes	Long	Cash won for single spin (in cents) without partner currency multiplier
creditsPlayed	Yes	Long	Credits played for single spin
creditsWon	Yes	Long	Credits won for single spin
freeSpin	Yes	Boolean	The status of spin type: "true" means the spin type is free game. "false" means the spin type is normal one.
round	Yes	Integer	Ordinal number of round for a single free spin session
roundsRemaining	Yes	Integer	Number of rounds remaining counting down from this round

BonusPickReport Parameter Structure

Parameter	**Required**	**Type**	**Usage**
playerId	**Yes**	String	Player ID shall be unique to represent a player
operator	No	String	Operator or agent name for the player if any
transactionId	Yes	String	The transaction Id to identify a spin. Free spin will share the same transaction Id with the dependent normal spin to identify as a "Coin Out"
gameName	Yes	String	The name of the game to spin
pickDate	No	String	The date time of the bonus pick is complete. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-21 15:56:18)
tzOffset	No	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, use -4, UTC+8 use 8, UTC+0, use 0)
currency	Yes	String	ISO 4217 currency code
denomination	Yes	Float	The unit of money to won
cashWon	Yes	Long	Cash won for a single bonus pick session (in cents)
cashWonRaw	Yes	Long	Cash won for a single bonus pick session (in cents) without partner currency multiplier
creditsWon	Yes	Long	Credits won for a single bonus pick session

7.3 Summary Report Request

This is sent from licensee to KAGA RMP to get a summary of all bets, wins, and profit/loss history for a given date range. Because this report is highly performant, no concerns need to be made on date ranges. At this time, only the UTC-4, UTC+0, and UTC+8 timezones are supported. If additional timezones are needed, please contact your account representative.

Content-type: application/json
URL: https://kagarmp_server/kaga/summaryReport?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
fromDate	Yes	String	The start date of the summary report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20)
toDate	Yes	String	The end date of the summary report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, please use -4, UTC+8 please use 8, UTC+0, please use 0)

Successful JSON Response

Parameter	Required	Type	Usage
status	Yes	String	A status string or message.
statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
fromDate	Yes	String	The start date of the summary report. This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
toDate	Yes	String	The end date of the summary report. This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0 (e.g. UTC-4, use -4, UTC+8 use 8, UTC+0, use 0)
summaries	No	Array	An array of SummaryLineItem(s). If no summaries exist for the given date range, this field will come back empty or null. Please see

			the table below for the structure of each SummaryLineItem object.

SummaryLineItem Parameter Structure

Parameter	Required	Type	Usage
currency	Yes	String	ISO 4217 currency code
numGames	Yes	Integer	The number of unique games played
totalSpins	Yes	Integer	The total number of plays. This is inclusive of all wagered spins as well as any free spins
wageredSpins	Yes	Integer	The total number of wagered/bet plays. This number only includes spins that the player made an actual bet on (coin-in)
freeSpins	Yes	Integer	The total number of free spins played
bonusGames	Yes	Integer	The total number of **non**-spin mini bonus games played
cashTurnover	Yes	Decimal	The unformatted, monetary amount of all bets for the given currency of this line item
cashWon	Yes	Decimal	The unformatted, monetary amount of all wins for the given currency of this line item
netGGR	Yes	Decimal	The unformatted, monetary amount of Gross Gaming Revenue (net-win) for the given currency of this line item
cashTurnOverStr	Yes	String	Formatted, monetary amount of all bets for the given currency of this line item
cashWonStr	Yes	String	Formatted, monetary amount of all wins for the given currency of his line item
netGGRStr	Yes	String	Formatted, monetary amount of Gross Gaming Revenue (net-win) for the given currency of this line item
hold	Yes	Decimal	The hold/win percentage (cashWon / cashTurnover) for the given line item
holdStr	Yes	String	The formatted percentage of hold
partnerBonusWager	No	Decimal	The total number of wagered/bet plays under the rewarded game (free play). This number only includes spins that the player made an actual bet on (coin-in)
partnerBonusCashWon	No	Decimal	The unformatted, monetary amount of all wins under the rewarded game (free play), and the rewarded game belongs to cash back

partnerBonusBonusWon	No	Decimal	The unformatted, monetary amount of all wins under the rewarded game (free game), and the rewarded game belongs to bonus
partnerBonusWagerStr	No	String	Formatted, monetary amount of all bets under the rewarded game (free play)
partnerBonusCashWonStr	No	String	Formatted, monetary amount of all wins under the rewarded game (free play), and the rewarded game belongs to cash back
partnerBonusBonusWonStr	No	String	Formatted, monetary amount of all wins under the rewarded game (free play), and the rewarded game belongs to bonus

7.4 Game Report Request

This is sent from licensee to KAGA RMP to get a comprehensive game spin and bonus pick history record report. This API uses pagination to optimize the performance.

Content-type: application/json
URL: https://kagarmp_server/kaga/gamePlayReport?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
currency	No	String	ISO 4217 currency code. If currency is not specified, it will return data related with all currency
fromDate	Yes	String	The start date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20 15:56:18)
toDate	Yes	String	The end date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-21 15:56:18)
tzOffset	No	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, please use -4, UTC+8 please use 8, UTC+0, please use 0) This is used to setup spinDate, pickDate with timezone. The default value is 0
offset	Yes	Long	The offset of the game report records. The offset of the initial record is 0
numOfRecords	Yes	Long	The number of the game report records to return. The maximum number is 10000 for each request
operatorName	No	String	An operator or agent name that the player belongs to. This field is optional and should only be used if the

			partner has agents or other operators. Note that passing the "master" keyword for operatorName will retrieve plays taken place under the partner with no designated operators. If this value is omitted/null, then all plays from all operators and master partner are retrieved.

Successful JSON Response

Parameter	Required	Type	Usage
status	Yes	String	A status string or message.
statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
fromDate	Yes	String	The start date of the game report . This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20 15:56:18)
toDate	Yes	String	The end date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20 15:56:18)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, use -4, UTC+8 use 8, TC+0, use 0)
offset	Yes	Long	The offset of the game report records
numOfRecords	Yes	Long	The number of the game report records to return
hasNext	Yes	Boolean	To specify if more game report records could be retrieved between fromDate and toDate parameter
spinReport	No	Array	Spin history including free spin and normal spin between fromDate and toDate parameter
bonusPickReport	No	Array	Bonus pick history between fromDate and toDate parameter

7.5 Player Game Report Request

This is sent from licensee to KAGA RMP to get one specific player's comprehensive game spin and bonus pick history record report. This will retrieve all of the players' bets and activity constrained by the dates and currency. This API uses pagination to optimize the performance.

Content-type: application/json
URL: https://kagarmp_server/kaga/playerGamePlayReport?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage

playerId	Yes	String	Player ID shall be unique to represent a player, It sent to the Game Launch URL
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
currency	No	String	ISO 4217 currency code. If currency is not specified, it will return data related with all currency
fromDate	Yes	String	The start date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20 15:56:18)
toDate	Yes	String	The end date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-21 15:56:18)
tzOffset	No	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, please use -4, UTC+8 please use 8, UTC+0, please use 0) This is used to setup spinDate, pickDate with timezone. The default value is 0
offset	Yes	Long	The offset of the game report records. The offset of the initial record is 0
numOfRecords	Yes	Long	The number of the game report records to return. The maximum number is 10000 for each request
operatorName	No	String	An operator or agent name that the player belongs to. This field is optional and should only be used if the partner has agents or other operators. Note that passing the “master” keyword for operatorName will retrieve plays taken place under the partner with no designated operators. If this value is omitted/null, then all plays from all operators and master partner are retrieved.

Successful JSON Response

Parameter	Required	Type	Usage
status	Yes	String	A status string or message.
statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
fromDate	Yes	String	The start date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20 15:56:18)
toDate	Yes	String	The end date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20 15:56:18)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, use -4, UTC+8 use 8, TC+0, use 0)

offset	Yes	Long	The offset of the game report records
numOfRecords	Yes	Long	The number of the game report records to return
hasNext	Yes	Boolean	To specify if more game report records could be retrieved between fromDate and toDate parameter
spinReport	No	Array	Spin history including free spin and normal spin between fromDate and toDate parameter
bonusPickReport	No	Array	Bonus pick history between fromDate and toDate parameter
lastUsedDate	No	String	The last date that player played with time zone UTC+0 (e.g. 2017-06-20 15:56:18.0)
lastDeviceOS	No	String	The device OS that player used (e.g. Windows)
lastDeviceType	No	String	The browser that player used (e.g. Chrome)
lastDeviceVersion	No	String	The browser version that player used (e.g. 61.0.3163.91)

7.6 Game Report Request (No Pagination – Not recommended for use)

This is sent from licensee to KAGA RMP to get a comprehensive game spin and bonus pick history record report. Please limit the time range on this report to limit the amount of data returned.

Content-type: application/json
URL: https://kagarmp_server/kaga/playerReport?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage
playerId	No	String	Player ID shall be unique to represent a player, It sent to the Game Launch URL. If playerId is not specified, it will return all the player related with the partnerName
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
randomId	Yes	Integer	A random number that changes for every game list call
currency	No	String	ISO 4217 currency code. If currency is not specified, it will return data related with all currency
from	Yes	String	The start date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20 15:56:18)

to	Yes	String	The end date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-21 15:56:18)
tzOffset	No	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, please use -4, UTC+8 please use 8, UTC+0, please use 0) This is used to setup spinDate, pickDate with timezone. The default value is 0
operatorName	No	String	An operator or agent name that the player belongs to. This field is optional and should only be used if the partner has agents or other operators. Note that passing the "master" keyword for operatorName will retrieve plays taken place under the partner with no designated operators. If this value is omitted/null, then all plays from all operators and master partner are retrieved.

Successful JSON Response

Parameter	Required	Type	Usage
status	Yes	String	A status string or message.
statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
from	Yes	String	The start date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20 15:56:18)
to	Yes	String	The end date of the game report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20 15:56:18)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, use -4, UTC+8 use 8, TC+0, use 0)
wageredSpinsPlayed	No	Long	Sum of played spin count
freeSpinsPlayed	No	Long	Sum of played free spin count
freePlayPlayed	No	Long	Sum of played free play count
totalReport	No	Array	Sum for cash played and cash won history by currency
spinReport	No	Array	Spin history including free spin and normal spin between "from" and "to" parameter
bonusPickReport	No	Array	Bonus pick history between "from" and "to" parameter

totalReport **Parameter Structure**

Parameter	Required	Type	Usage
currency	Yes	String	ISO 4217 currency code

totalCashPlayed	Yes	String	Sum of cash played between "from" and "to" parameter. This field is formatted to the currency type and symbol
totalMoneyWon	Yes	String	Sum of money won between "from" and "to" parameter. This field is formatted to the currency type and symbol

7.7 Game Summary Report Request

This is sent from licensee to KAGA RMP to get the specific game(s) summary of all bets, wins, and profit/loss history for a given date range. At this time, only the UTC-4, UTC+0, and UTC+8 timezones are supported. If additional timezones are needed, please contact your account representative.

Content-type: application/json
URL: https://kagarmp_server/kaga/gameSummaryReport?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
fromDate	Yes	String	The start date of the game summary report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20)
toDate	Yes	String	The end date of the game summary report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, please use -4, UTC+8 please use 8, UTC+0, please use 0)
summaryType	Yes	String	A string indicating what kind of game summary will be generated. Currently, "gameByDay" and "gameByGame" are supported.
gameFilter	No	String	A list of game name indicating only listed game(s) summary should be return. (e.g. "SuperShot,Fantasy777")
offset	Yes	Long	The offset of the game summary report records. The offset of the initial record is 0
numOfRecords	Yes	Long	The number of the game summary report records to return. The maximum number is 10000 for each request

Successful JSON Response

Parameter	Required	Type	Usage
status	Yes	String	A status string or message.
statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
fromDate	Yes	String	The start date of the game summary report. This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
toDate	Yes	String	The end date of the game summary report. This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0 (e.g. UTC-4, use -4, UTC+8 use 8, UTC+0, use 0)
offset	Yes	Long	The offset of the game summary report records
numOfRecords	Yes	Long	The number of the game summary report records to return
hasNext	Yes	Boolean	To specify if more game summary report records could be retrieved between fromDate and toDate parameter
summaries	No	Array	An array of SummaryLineItem(s). If no summaries exist for the given date range, this field will come back empty or null. Please see the table below for the structure of each SummaryLineItem object.

SummaryLineItem Parameter Structure

Parameter	Required	Type	Usage
fromDate	Yes	String	The start date of one specific game summary report. This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
toDate	Yes	String	The end date of one specific game summary report. This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0 (e.g. UTC-4, use -4, UTC+8 use 8, UTC+0, use 0)
gameName	Yes	String	Unique name or id of the game
gameType	Yes	String	The game type. Valid values are (“slots”, “fish”, “vpoker”, “pvp”, “other”, “table”)
currency	Yes	String	ISO 4217 currency code

numGames	Yes	Integer	The number of unique games played
totalSpins	Yes	Integer	The total number of plays. This is inclusive of all wagered spins as well as any free spins
wageredSpins	Yes	Integer	The total number of wagered/bet plays. This number only includes spins that the player made an actual bet on (coin-in)
freeSpins	Yes	Integer	The total number of free spins played
bonusGames	Yes	Integer	The total number of non-spin mini bonus games played
cashTurnover	Yes	Decimal	The unformatted, monetary amount of all bets for the given currency of this line item
cashWon	Yes	Decimal	The unformatted, monetary amount of all wins for the given currency of this line item
netGGR	Yes	Decimal	The unformatted, monetary amount of Gross Gaming Revenue (net-win) for the given currency of this line item
cashTurnOverStr	Yes	String	Formatted, monetary amount of all bets for the given currency of this line item
cashWonStr	Yes	String	Formatted, monetary amount of all wins for the given currency of his line item
netGGRStr	Yes	String	Formatted, monetary amount of Gross Gaming Revenue (net-win) for the given currency of this line item
hold	Yes	Decimal	The hold/win percentage (cashWon / cashTurnover) for the given line item
holdStr	Yes	String	The formatted percentage of hold
partnerBonusWager	No	Decimal	The total number of wagered/bet plays under the rewarded game (free play). This number only includes spins that the player made an actual bet on (coin-in)
partnerBonusCashWon	No	Decimal	The unformatted, monetary amount of all wins under the rewarded game (free play), and the rewarded game belongs to cash back
partnerBonusBonusWon	No	Decimal	The unformatted, monetary amount of all wins under the rewarded game (free game), and the rewarded game belongs to bonus
partnerBonusWagerStr	No	String	Formatted, monetary amount of all bets under the rewarded game (free play)

partnerBonusCashWonStr	No	String	Formatted, monetary amount of all wins under the rewarded game (free play), and the rewarded game belongs to cash back
partnerBonusBonusWonStr	No	String	Formatted, monetary amount of all wins under the rewarded game (free play), and the rewarded game belongs to bonus

7.8 Player Game Summary Report Request

This is sent from licensee to KAGA RMP to get the specific game(s) summary of all bets, wins, and profit/loss history for a given date range and for a given set of players. This report only contains daily summaries for players/games and does not contain details about the individual bets. At this time, only the UTC-4, UTC+0, and UTC+8 timezones are supported. If additional timezones are needed, please contact your account representative.

Content-type: application/json
URL: https://kagarmp_server/kaga/playerGameSummaryReport?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
fromDate	Yes	String	The start date of the game summary report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20)
toDate	Yes	String	The end date of the game summary report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, please use -4, UTC+8 please use 8, UTC+0, please use 0)
gameFilter	No	String array	A list of game name indicating only listed game(s) summary should be return. If this is empty, all played games are returned. (e.g. ["SuperShot","Fantasy777"])
playerFilter	No	String array	A list of playerId's for which the summary should return. If this is empty, data for all players are returned. (e.g. ["Player1","Player2"])
operatorFilter	No	String array	A list of operator names for which the summary should return. If this is empty, data for all operators are returned (e.g. ["OperatorA","OperatorB"])

Successful JSON Response

Parameter	Required	Type	Usage
status	Yes	String	A status string or message.
statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
fromDate	Yes	String	The start date of the game summary report. This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
toDate	Yes	String	The end date of the game summary report. This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0 (e.g. UTC-4, use -4, UTC+8 use 8, UTC+0, use 0)
summaries	No	Array	An array of PlayerGameSummaryLineItem(s). If no summaries exist for the given date range, this field will come back empty or null. Please see the table below for the structure of each PlayerGameSummaryLineItem object.

PlayerGameSummaryLineItem Parameter Structure

Parameter	Required	Type	Usage
date	Yes	String	The date . This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0 (e.g. UTC-4, use -4, UTC+8 use 8, UTC+0, use 0)
playerName	Yes	String	The unique player name or player ID
operatorName	No	String	The operator to which this player belongs.
gameName	Yes	String	Unique name or id of the game
gameType	Yes	String	The game type. Valid values are (“slots”, “fish”, “vpoker”, “pvp”, “other”, “table”)
currency	Yes	String	ISO 4217 currency code
numWagers	Yes	Integer	The total number of wagered/bet plays. This

			number only includes spins that the player made an actual bet on (coin-in)
numFreeGames	Yes	Integer	The total number of free games played
numBonusGames	Yes	Integer	The total number of non-spin mini bonus games played
totalBet	Yes	Decimal	The unformatted, monetary amount of all bets for the given currency of this line item
totalBaseGamesWin	Yes	Decimal	The unformatted, monetary amount of all wins that are not from mini bonus games or free games for the given currency of this line item.
totalFreeGamesWin	Yes	Decimal	The unformatted, monetary amount of all free game wins.
totalBonusGamesWin	Yes	Decimal	The unformatted, monetary amount of all mini-bonus game wins.
totalWin	Yes	Decimal	The unformatted, monetary amount of all wins. Note this value is a sum of totalFreeGamesWin, totalBonusGamesWin, and totalBaseGamesWin.
bonusGamesIncomplete	Yes	Boolean	True if there are any mini bonus games that are incomplete and waiting for player interaction. False if all related mini bonus games are complete
partnerBonusTotalBet	No	Decimal	The unformatted, monetary amount of all bets under the rewarded game (free play)
totalBaseGamesPartnerBonusCashWin	No	Decimal	The unformatted, monetary amount of all wins that are not from mini bonus games or free games under the rewarded game (free play), and the rewarded game belongs to cash back

totalFreeGamesPartnerBonusCashWin	No	Decimal	The unformatted, monetary amount of all free game wins under the rewarded game, and the rewarded game belongs to cash back
totalBaseGamesPartnerBonusBonusWin	No	Decimal	The unformatted, monetary amount of all wins that are not from mini bonus games or free games under the rewarded game (free play), and the rewarded game belongs to bonus
totalFreeGamesPartnerBonusBonusWin	No	Decimal	The unformatted, monetary amount of all free game wins under the rewarded game, and the rewarded game belongs to bonus

7.9 Daily Game Summary Report Request

This is sent from licensee to KAGA RMP to get each game summary of all bets, wins, and profit/loss history for a given date. The summary is in hours. This report only contains hourly summaries for games and does not contain details about the individual bets. At this time, only the UTC-4, UTC+0, and UTC+8 timezones are supported. If additional timezones are needed, please contact your account representative.

Content-type: application/json
URL: https://kagarmp_server/kaga/dailyHourSummaryReport?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
date	Yes	String	The date of the game summary report. This date should coincide with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0. (e.g. UTC-4, please use -4, UTC+8 please use 8, UTC+0, please use 0)

Successful JSON Response

Parameter	Required	Type	Usage
status	Yes	String	A status string or message.

statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
date	Yes	String	The date of the game summary report. This date coincides with the timezone specified in tzOffset (e.g. 2017-06-20)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0 (e.g. UTC-4, use -4, UTC+8 use 8, UTC+0, use 0)
summaries	No	Array	An array of DailyHourSummaryLineItem(s). If no summaries exist for the given date, this field will come back empty or null. Please see the table below for the structure of each DailyHourSummaryLineItem object.

DailyHourSummaryLineItem Parameter Structure

Parameter	Required	Type	Usage
datehour	Yes	String	The datetime in hours. This datetime coincides with the timezone specified in tzOffset (e.g. 2017-06-20 05:00:00)
tzOffset	Yes	Integer	The timezone offset in hours from UTC+0 (e.g. UTC-4, use -4, UTC+8 use 8, UTC+0, use 0)
gameName	Yes	String	Unique name or id of the game
gameType	Yes	String	The game type. Valid values are (“slots”, “fish”, “vpoker”, “pvp”, “other”, “table”)
currency	Yes	String	ISO 4217 currency code
numWagers	Yes	Integer	The total number of wagered/bet plays. This number only includes spins that the player made an actual bet on (coin-in)
numFreeGames	Yes	Integer	The total number of free games played
numBonusGames	Yes	Integer	The total number of non-spin mini bonus games played
totalBet	Yes	Decimal	The unformatted, monetary amount of all bets for the

			given currency of this line item
totalBaseGamesWin	Yes	Decimal	The unformatted, monetary amount of all wins that are not from mini bonus games or free games for the given currency of this line item.
totalFreeGamesWin	Yes	Decimal	The unformatted, monetary amount of all free game wins.
totalBonusGamesWin	Yes	Decimal	The unformatted, monetary amount of all mini-bonus game wins.
totalWin	Yes	Decimal	The unformatted, monetary amount of all wins. Note this value is a sum of totalFreeGamesWin, totalBonusGamesWin, and totalBaseGamesWin.
bonusGamesIncomplete	Yes	Boolean	True if there are any mini bonus games that are incomplete and waiting for player interaction. False if all related mini bonus games are complete
partnerBonusTotalBet	No	Decimal	The unformatted, monetary amount of all bets under the rewarded game (free play)
totalBaseGamesPartnerBonusCashWin	No	Decimal	The unformatted, monetary amount of all wins that are not from mini bonus games or free games under the rewarded game (free play), and the rewarded game belongs to cash back
totalFreeGamesPartnerBonusCashWin	No	Decimal	The unformatted, monetary amount of all free game wins under the rewarded game, and the rewarded game belongs to cash back
totalBaseGamesPartnerBonusBonusWin	No	Decimal	The unformatted, monetary amount of all wins that are not from mini bonus games or free games under the

			rewarded game (free play), and the rewarded game belongs to bonus
totalFreeGamesPartnerBonusBonusWin	No	Decimal	The unformatted, monetary amount of all free game wins under the rewarded game, and the rewarded game belongs to bonus

KA Gaming

8. JACKPOT

8.1 Partner Progressive Request

This is sent from licensee to KAGA RMP to get the current Jackpot status of a game.

Content-type: application/json
URL: https://kagarmp_server/kaga/partnerProgressiveList?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
gameId	Yes	String	Unique name or id of the game
currency	Yes	String	ISO 4217 currency code

Successful JSON Response

Parameter	Required	Type	Usage
status	Yes	String	A status string or message.
statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
partnerProgressive	No	Array	An array of PartnerProgressiveItem(s). If no progressive exists for the given game, this field will come back empty or null. Please see the table below for the structure of each PartnerProgressiveItem object.

PartnerProgressiveItem Parameter Structure

Parameter	Required	Type	Usage
gameId	Yes	String	Unique name or id of the game
currency	Yes	String	ISO 4217 currency code
progressiveName	Yes	String	The Jackpot name of the game
currentHitValue	Yes	Long	The Jackpot hit value of the game in cents
currentValue	Yes	Float	The Jackpot current value of the game in cents

KA Gaming

9. PROMOTION

9.1 Promotion Info Request

This is sent from licensee to KA's RMP to get information about previous or ongoing promotion events.

KA supports three different types of promotions:

1. TURNOVER – Qualifying players of TURNOVER type promotions are ranked based on the amount of qualified total bet played.
2. WINS – Qualifying players of WINS type promotions are ranked based on the qualified total win amount won by the player.
3. WAGERS – Qualifying players of WAGERS type promotions are ranked based on the qualified number of individual wagers played by the player.

Content-type: application/json
URL: https://kagarmp_server/kaga/promotionInfo?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
currency	No	String	ISO 4217 currency code. If currency is not specified, it will return data related with all currency
startDate	No	String	The start date range the returned promotions should fall under. If not specified will return all promotions starting from any time and within the other filtering parameters sent in the request. (Note timezone is UTC+0 e.g. 2017-06-20 15:56:18)
endDate	No	String	The end date range the returned promotions should fall under. If not specified will return all promotions ending at any time and within the other filtering parameters sent in the request. (Note timezone is UTC+0 e.g. 2017-06-21 15:56:18)
promotionId	No	String	A unique promotion name. If specified, only the specific promotion corresponding to the given promotion Id will be in the response

Successful JSON Response

Parameter	Required	Type	Usage
status	Yes	String	A status string or message.

statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
promotions	No	Array	Array of Promotion objects matching the request criteria. (See below for Promotion object format)

Promotion Parameter Structure

Parameter	Required	Type	Usage
promotionId	Yes	String	Unique ID of the promotion
startDate	Yes	String	The start date for the promotion. Formatted as 2019-01-01 13:45:10 with timezone as UTC+0
endDate	Yes	String	The end date for the promotion. Formatted as 2019-01-01 13:45:10 with timezone as UTC+0
ended	Yes	Boolean	True indicates promotion has ended. False indicates promotion is still active or has not yet started.
paid	Yes	Boolean	True indicates promotion winners have been paid. False indicates promotion winners have yet to be paid.
currency	Yes	String	The currency of players, plays, and prizes for this promotion. ISO4217 currency codes are used
type	Yes	String	The promotion type. Valid types are 'TURNOVER', 'WINS', 'WAGERS'.
games	No	Array	An array of strings for the games qualified for this promotion. If this value is null or empty, that means all games qualify
totalPrizeValue	Yes	Decimal	The total prize values for all winning ranks within the promotion
leaderboard	No	Array	Array of PromotionLeaderboardRank objects. Each object in the array represents a current ranking within the promotion. Only players that have qualified for one of the promotion ranks are returned. See below for details of PromotionLeaderboardRank objects.

PromotionLeaderboardRank Parameter Structure

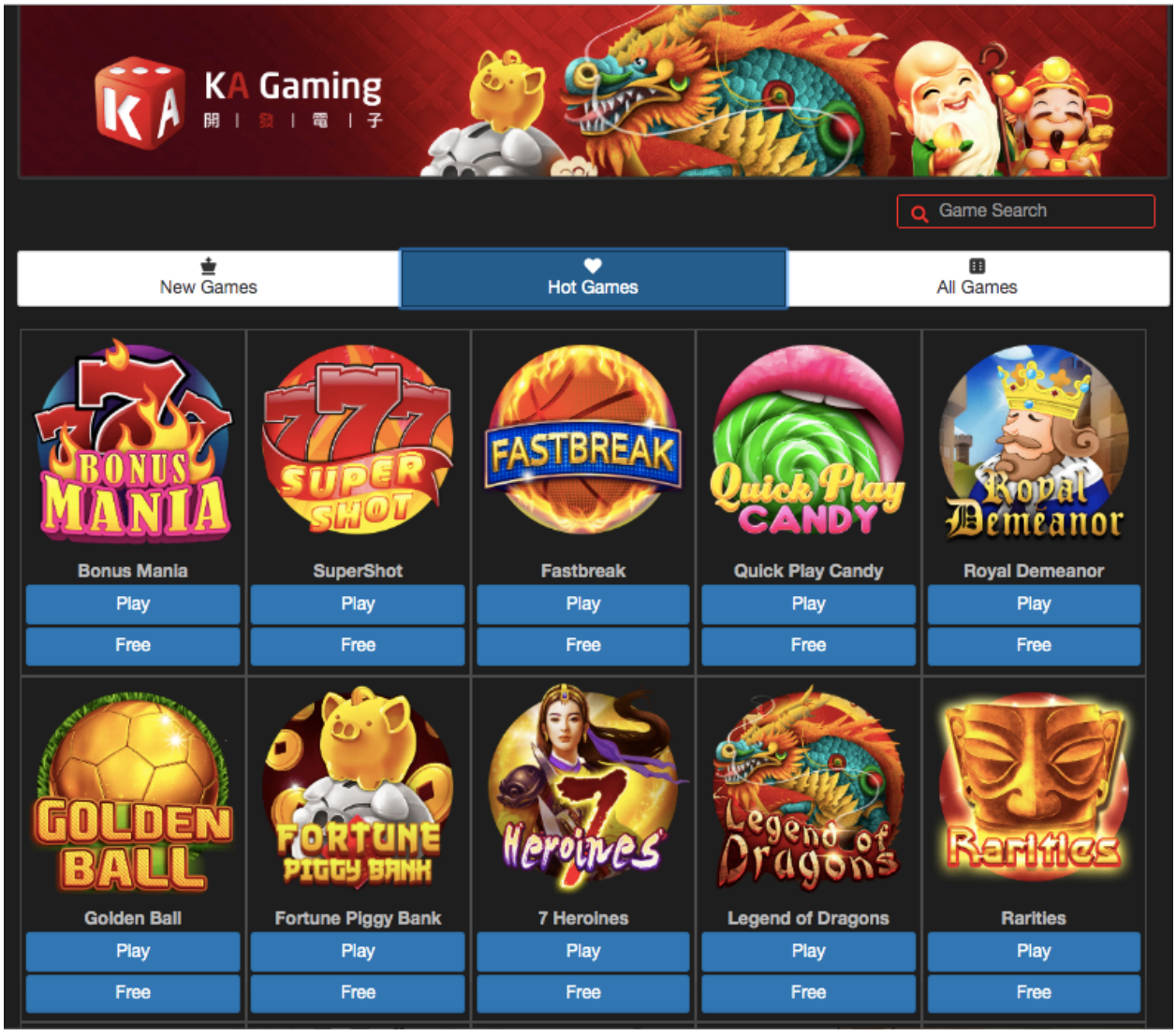
Parameter	Required	Type	Usage
rank	Yes	Integer	The win rank for the promotion player
playerName	Yes	String	The unique player name or player ID for the ranking promotion player
score	Yes	Decimal	For turnover based promotions, the total qualified amount (in currency)

			wagered by the player during the promotion. For wins based promotions, the total qualified win amount (in currency) won by the player during the promotion. For wagers based promotions, the total number of qualified wagers by the player during the promotion
prize	Yes	Decimal	The amount won by the player in the promotion (in currency)
paymentWalletTransactionId	No	String	The wallet transaction ID for the payment made to the player for winning a prize in the promotion. If the player has not been paid, this field will be null.
paymentDateUTC0	No	String	The date promotion prize was paid. The date is in yyyy-MM-dd HH:mm:ss format and is in the UTC+0 timezone

10. KA LOBBY

10.1 Lobby Usage

KA offers a pre-made lobby that can be conveniently linked to start games for the players. It requires that all backend wallet (seamless or transfer) API integrations are complete and functional.



The lobby can be opened by formulating the lobby URL (see table below regarding parameters): https://lobby.kaga88.com/?tp=<ICON_TYPE>&loc=<LANGUAGE>&nw=<OPEN_GAMES_IN_NEW_WINDOW>&p=<PARTNER_ID>&ak=<ACCESS_KEY>&u=<USER_ID>&cr=<CURRENCY>&o=<OPERATOR_NAME>&t=<TOKEN>

Lobby URL Parameters

Parameter	Required	Type	Usage
p	Yes	String	Partner ID – same partnerID used in Wallet API's and provided by your KA representative
ak	Yes	String	Access Key – same access key used in Wallet API's and provided by your KA representative
u	Yes	String	Unique Identifier for the player. This can be the player's username, or any other persistent and consistent value used by the licensee to identify the player with RMP
cr	Yes	String	Currency to use for the player. Please use ISO 4217 currency codes
t	No	String	Unique token string generated by licensee that is unique for each Game Launcher link. The token will be sent back by RMP to the licensee during the "start" REST service and should be used when received by the licensee to verify that the player did launch the game
nw	No	Integer	nw=1 will open games in a new browser window. nw=0 will open games in the existing browser window. If unspecified, the default behavior is to open games in the existing browser
o	No	String	The licensee identifier for sub-licensees or agents under the licensee for which this player or the intended games played should be accounted under
tp	No	String	The icon type to show. Valid values are: - circular - rectangular - square - circular_framed - oval - oval_framed

KA Gaming

11. CRYPTO CURRENCY SUPPORT

11.1 Crypto Currency Support & Usage

KA supports several crypto currencies with no additional customizations needed. These currencies are:

Crypto Currency	Currency Code	Multiplier	Currency Display
Bitcoin	BTC	1 / 1000	mBTC
Ethereum	ETC	1 / 1000	mETC
LiteCoin	LTC	1 / 1000	mLTC
XMR	XMR	1 / 1000	mXMR
DASH	DAS	1 / 1000	mDASH
EOS	EOS	1 / 1000	mEOS
TRON	TRX	1	TRX
USDT	UST	1	USDT
USDC	USC	1	USC

Usage of each of these crypto currencies may include a multiplier to reduce the number of needed decimals needed to display a reasonable value to the player. For example, BTC uses a 1 / 1000 multiplier with two decimal places and is marked as mBTC. 1000 mBTC = 1 BTC, 1000 mEOS = 1 EOS, etc…

All functionalities of the KA RMP system are able to support the above crypto currencies the same as any other ISO4217 currency. To use these crypto currencies, use the correct Currency Code in the game launch URL, lobby URL, and any of the promotion, seamless wallet or detached wallet API's.

An example game launch URL (example using mBTC):
https://<KAGASERVER>/?g=<GAME_NAME>&p=<PARTNER_NAME>&u=<UNIQUE_PLAYERID>&t=<UNIQUE_TOKEN>&ak=<PARTNER_ACCESSKEY>&loc=<PLAYER_LANGUAGE>&cr=BTC&m=<GAME_MODE>&dn=<DENOMINATION>&o=<OPERATOR>&sn=<SOUND_OFF>&l=<LOBBY_URL>&sl=<SUPPORT_URL>&da=<DISPLAY_NAME>&ro=<REMOVE_OPTIONS>&asm=<AUTOSPIN_MODE>&if=<IFRAME_MODE>&mc=<BALANCE_METER_MODE>&v=<VERTICAL_MODE>&tl=<TITLE>&ld=<LOW_DEFINITION>&dc=<DISABLE_EXIT_CONFIRMATION&db=<DISABLE_INSUFFICIENT_BALANCE_ALERT>

An example lobby URL (example using TRX):
https://lobby.kaga88.com/?tp=<ICON_TYPE>&loc=<LANGUAGE>&nw=<OPEN_GAMES_IN_NEW_WINDOW>&p=<PARTNER_ID>&ak=<ACCESS_KEY>&u=<USER_ID>&cr=TRX&o=<OPERATOR_NAME>&t=<TOKEN>

An example seamless wallet start game request (example using EOS):
{.

“timestamp”: 123456901,
“sessionId”: “aefdc139501faabcddaf”,
“partnerPlayerId”: “player1”,
“currency”: “EOS”,
“gameId”: “BonusMania”,
“token”: “token”
“action”: “start”,
“playerIp”: “211.72.236.175”
}

An example detached wallet deposit request (example using ETC):
{
“username”: “player1”,
“partnerName”: “partnerID”,
“currency”: “ETC”,
“accessKey”: “myaccesskey”,
“depositAmount”, 100000 (this is equivalent to 1000.00 mETC = 1 ETC)
}

KA Gaming

12. Game Launch Token

12.1 Create Launch Token Request

This is sent from licensee to KA's RMP to generate game launch token if licensee enables single used game launch token validation feature. Enabling single use game launch token for web games prevents game launch URL's from being re-used. The same can also be accomplished in seamless wallets by partners' own token validation.

The Game Launch Token API implementation is required for the KA Real Money Native App and with the "app" POST JSON parameter set as true. The KA Real Money Native App uses the Game Launch Token to verify valid app launch and associate app session with the correct real money wallet.

Content-type: application/json
URL: https://kagarmp_server/kaga/createLaunchToken?hash=<HMAC>

POST JSON Parameters:

Parameter	Required	Type	Usage
partnerName	Yes	String	The unique name of the licensee
accessKey	Yes	String	The access key for the licensee
operatorName	No	String	The operator, agent, sub-agent, or brand used to identify players
currency	Yes	String	ISO 4217 currency code
username	Yes	String	A unique player identifier used by the licensee
App	No	Boolean	Indicate if created launch token is used for App launch or web launch. If omitted, the default is web launch

Successful JSON Response

Parameter	Required	Type	Usage
Status	Yes	String	A status string or message
statusCode	Yes	Integer	A shared numerical error code. Please refer to the section on status codes to see which are available and their usage
username	Yes	String	A unique player identifier used by the licensee
currency	Yes	String	ISO 4217 currency code
launchToken	Yes	String	Unique token string generated by KA's RMP that is unique for each game launcher link. The token will be valid for single used and expired after game launched

KA Gaming

13. Licensee Free/Rewarded Games

13.1 Licensee Free/Rewarded Games

For seamless wallet, KA's RMP will send play request to licensee to validate the wager/bet, licensee could set a flag to indicate this play is a rewarded play (free play) on successful response.

In Section 5.8 Play Request, KA's RMP adds more partnerBonusPlay parameter for licensee to indicate the current play is a rewarded game (free play).

More parameters for licensee rewarded games (5.8 Play Response)

Parameter	Required	Type	Usage
partnerBonusPlay	No	Int	To indicate the wager is a rewarded game (free play), and two types are supported: 1: The rewarded belongs to cash back 2: The rewarded belongs to bonus
partnerBonusBalance	No	Long	The player's current rewarded balance remaining, in cents

When licensee sets the partnerBonusPlay parameter to indicate the play is a rewarded game (free play), some of the reporting API will generate the summary based on this information, Those APIs are listed below, for more detail description please refer to the Chapter 7:

- 7.3 Summary Report Request
- 7.7 Game Summary Report Request
- 7.8 Player Game Summary Report Request
- 7.9 Daily Game Summary Report Request

14. Real Money Native App

14.1 Real Money Native App Usage

KA offers a real money native mobile app versions on iOS and Android of all available KA games and game types in the KA games library. Being a true native app, the native KA games and game types offer better performance, graphics, sound quality, and player immersion compared to the HTML5 versions of KA games. The native app utilizes the same integration, configuration and wallets as the HTML5 version of KA games. An additional integration of the Launch Token API as detailed in 12.1 of this document is required to use the KA Real Money Native App. The KA Real Money Native App works with both seamless and transfer wallet integrations.

14.2 Real Money Native App URL Creation

The KA Real Money Native App can be launched by using the custom KA Real Money Native App URL. As part of the URL, additional configuration parameters can be passed in to authenticate the player with the correct wallet as well as set other configuration parameters.

The custom KA app URL is formatted as:
https://<KAGASERVERURL>/applaunch?<parameterName1>=<parameterValue1>&<parameterName2>=<parameterValue2>

An example KA app URL:
https://games.kaga88.com/applaunch?token=f8cd62ea1456b2f549fb0b21db9ca612

Parameter	Required	Usage
token	Yes	This is the token created using the API detailed in Section 12.1. This token is used to link the app's session back to the correct user and wallet.
gameId	No	The game ID to start immediately upon app launch. If the gameID is not specified, the app will launch into the lobby with a selection of all enabled KA games and game types. If the gameID is specified, the app will launch directly into the specified game.
welcomeUrl	No	When specified, the game will open the welcomeUrl immediately within a pop-up menu of the app after successful app launch. The welcomeUrl needs to be URL encoded. The contents of the welcomeUrl can also be re-opened by the player from the game settings menu within the app. The welcomeUrl can be used for informational and branding of the KA native app. The "token" used to open the KA native app will be appended as a GET/Query parameter named "token" to the end of the welcomeUrl. This token can be used to identify the player and wallet and customize the contents of the welcomeUrl to the player.
exitUrl	No	When specified, on player exit of the KA native app, the player will be sent back to the exitUrl. The exitUrl needs to be URL encoded. The "token" used to open the KA native app will be appended as a GET/Query parameter named "token" to the end of the exitUrl. This token can be used to identify the player and wallet and customize the contents of the exitUrl to the player.
cashierUrl	No	When specified, a cashier option becomes available in the settings menus of the KA native app that opens the contents of the cashierUrl in a popup menu within the app. The cashierUrl needs to be URL encoded. The intent of the

		cashierUrl is for the player to deposit, withdraw, or transfer from their partner wallet balance hosted by the partner. The "token" used to open the KA native app will be appended as a GET/Query parameter named "token" to the end of the cashierUrl. This token can be used to identify the player and wallet and customize the contents of the cashierUrl to the player.

14.3 Real Money Native App Launch Flow

The general flow of launch the KA real money native app is the following:

1. Player logs in to partner site from an Android or iOS device and operates his wallet to ensure proper money balance in wallet.
2. Player presented with option to launch the KA native app into either the lobby or directly into a specific game.
3. Player presses "Play In Native App" or other similar button.
4. Partner site calls the Launch Token API (see section 12.1 of this document) and generates a Launch Token with the "app" option specified when calling this API.
5. After obtaining the app token, a KA Real Money Native App launch URL is created by the partner (see section 14.2 of this document)
6. The KA Real Money Native App redirects or opens a new tab/window with the generated KA Real Money Native App launch URL.
7. If the player has previously downloaded and installed the KA native app, the KA native app is directly opened. If the player has not yet installed the KA Real Money native app, he will be directed to download the appropriate app for iOS or Android.