



Integration API specification

Latest update: November, 2020

API version 3.131

Table of Contents

Version Control	7
I. API Overview	11
1.1 Seamless Wallet API	11
1.2 Balance Transfer API.....	12
1.3 Integration API	12
1.4 Data feeds.....	12
1.5 Player ID.....	13
Seamless Wallet (Authenticate method)	13
Balance Transfer (StartGame method)	13
1.6 Play session	13
1.7 Transaction reference.....	13
II. Integration API	14
2.1 GetCasinoGames	14
2.1.1 Game icons	17
2.2 DownloadStatistics	18
2.3 CloseSession.....	19
2.4 CancelRound	20
2.5 HealthCheck	21
III. Seamless Wallet API	22
3.1 Game opening.....	22
Explicit launch URL.....	22
GameLib JS function.....	22
3.2 Hash calculation	24
3.3 Data types.....	25
3.4 Authenticate	27
3.5 Authenticate2.....	29
3.6 Balance	30
3.7 Bet	31
3.8 Result.....	33
3.9 BonusWin.....	35
3.10 JackpotWin.....	37
3.11 EndRound.....	39
3.12 Refund	41
3.13 Withdraw	43
3.14 GetBalancePerGame.....	44

3.15	PromoWin.....	45
3.16	SessionExpired	47
3.17	Error codes.....	48
IV.	Balance Transfer API.....	49
4.1	Game opening.....	49
4.2	Hash calculation	49
4.3	Data types.....	50
4.4	CreatePlayer.....	51
4.5	Transfer.....	52
4.6	GetTransferStatus.....	54
4.7	GetBalance	55
4.8	StartGame.....	56
4.9	TerminateSession	57
4.10	Error codes.....	58
V.	Simple Free Rounds API	59
5.1	Hash calculation	59
5.2	Data types.....	59
5.3	CreateFRB	60
5.4	CancelFRB	61
5.5	GetPlayersFRB.....	62
5.6	Error codes.....	64
VI.	Variable Free Rounds API.....	65
6.1	Hash calculation	65
6.2	Data types.....	66
6.3	Create Free Rounds	66
6.4	Cancel Free Rounds	68
6.5	Add players	69
6.6	Remove players.....	70
6.7	Create Player Free Rounds.....	71
6.8	Error codes.....	73
VII.	Game History API.....	74
7.1	Hash calculation	74
7.2	Data types.....	75
7.3	GetPlayedGames	76
7.4	GetGameRounds	77
7.5	OpenHistory	79
7.6	GetRoundStatus	80

7.7	OpenHistoryExtended.....	81
7.8	Error codes.....	82
VIII.	Data feeds and reports	83
	Time points.....	83
	Using timepoints for past periods.....	83
	Data Feed URL	84
	Data formats.....	84
8.1	Data types.....	85
8.2	Game Rounds	87
8.3	In-game transactions.....	89
8.4	Failed Transactions.....	91
8.5	Active Jackpots.....	92
8.6	Jackpot Winnings.....	93
8.7	Daily Totals.....	94
8.8	Player's Incomplete Rounds.....	96
IX.	Tournament API	97
9.1	Tournament Winners.....	97
X.	Business Statistics API.....	99
	Hash calculation.....	99
10.1	Players daily report.....	99
10.2	Error codes.....	100
XI.	Diagrams	101
11.1	Flow diagram of opening casino game for Seamless Wallet API	101
11.2	Flow diagram of playing casino game Seamless Wallet API	102
XII.	Reconciliation	104
12.1	Reconciliation process	104
	Bet transaction	104
	Refund transaction	104
	Winning transactions	104
	End round.....	104
12.2	Reconciliation mechanism	104
XIII.	Reality Check	105
13.1	Server side implementation	105
	Reality Check Warning	105
	Reality Check Response.....	106
13.2	Client side implementation	107
XIV.	Custom messages	108

14.1	Custom messages syntax	108
14.2	Action types	109
14.3	Link types	109
14.4	Custom messages API	110
XV.	Regulated Markets	111
15.1	Responsible gaming - Sweden.....	111
	Login time.....	111
	Links and logos to self-assessment tests.....	111
	Set player's country	111
	Displaying of totals	111
15.2	Checksums API	111
15.3	Portuguese Regulated Market - Slot Machine Reporting	113
	Introduction	113
	Slot Machine Reporting API.....	113
	API specification	114
	Error codes.....	115
	Examples	115
XVI.	Integration with demo games	116
XVII.	Bingo Integration API	118
	Bingo Game API	118
	Definitions	118
	List of definitions for the OPERATOR:	118
17.1	Bingo room Launching.....	118
	Room opening	118
	Mini-games.....	121
17.2	Bingo integration API for OPERATOR side.....	122
	Nickname	122
	Free Round Bonus Prizes.....	123
17.3	Bingo integration API for PROVIDER side.....	124
	Hash calculation.....	124
	RoomList	125
	AvailableGames	127
	PreorderCards	129
	AvailableSessions.....	131
	SessionGameInfo	133
	BuySession.....	135
	PreorderedReport.....	136

JackpotWinners	138
ChangeNickname	140
Bingo Player History	141
Bingo Round History	143
Error Codes	145
17.4 Free tickets API	147
ActiveGameConfigs	147
CreateFreeTickets	149
CancelFreeTickets	150
ActiveFreeTickets	151
XVIII. Live Casino Dynamic Games Access	153
18.1 DGA Overview	153
18.2 WebSocket data feed	153
18.3 Data feeds	153
18.4 JavascriptAPI	154
18.5 Data Format	155
XIX. Games JavaScript API	159
19.1 Game Events Interception	159
19.2 Events	160
19.3 Triggers	160
Stop Autoplay	160
Update balance	160
19.4 Live Casino Games Full Screen JS	161

Version Control

Date	Author	Version	Changes description
22-Jun-16	Iryna Sh.	3.33	Added "Reconciliation" section.
10-Aug-16	Alex N.	3.40	Change document structure. Remove SOAP protocol. Add request parameters status field. Remove obsolete diagrams.
12-Sep-16	Iryna Sh.	3.41	Added Data Feeds -> Transactions.
27-Sep-16	Alex F.	3.42	Change document structure (footer, logo)
11-Oct-16	Iryna Sh.	3.43	Added Reality check chapter.
21-Oct-16	Alex F.	3.44	Added Integration with demo games chapter.
31-Oct-16	Alex B.	3.45	Added 1. Balance Transfer error codes 2. JS description for lobbyURL Seamless Wallet error codes moved under Seamless Wallet section
21-Dec-16	Alex B.	3.46	Seamless Wallet error codes: 120 error code has been added Minor fixes of idempotency / refund description
26-Dec-16	Alex F.	3.47	Added API requests
01-Feb-17	Alex F.	3.48	Add platform parameter for JackpotWin Add warnings regarding services protection at Pragmatic Play's side
9-Mar-17	Iryna Sh.	3.49	New method History API -> RoundStatus has been added.
21-Mar-17	Iryna Sh.	3.50	Status "Closed" in History http service -> Round Status was renamed to "Cancelled".
22-Mar-17	Alex N.	3.51	Update Free Rounds Bonus API Error codes
21-Apr-17	Alex B.	3.52	RoundStatus renamed to GetRoundStatus
27-Apr-17	Iryna Sh.	3.53	GetBalancePerGame method has been added to Seamless Wallet API. The "currency" field was removed from GetRoundStatus response.
28-Apr-17	Iryna Sh.	3.54	CloseSession method has been added to Integration API.
04-May-17	Alex N.	3.55	Add Error 8 to Balance Transfer API Error codes
12-May-17	Iryna Sh.	3.56	Statuses "CompleteInProgress" and "CancelInProgress" have been added to GetRoundStatus method in History API.
19-May-17	Alex N.	3.57	userId parameter is case-sensitive. Reality check response: request/response examples.
23-Jun-17	Iryna Sh.	3.58	1. FRB simple service has been renamed to Simple Free Rounds API 2. New Variable Free Rounds API has been added. 3. The document was slightly reformatted.
11-Jul-17	Alex B.	3.59+ zh	Optional token parameter added into Seamless Wallet API methods
19-Jul-17	Iryna Sh.	3.60	Reality check chapter has been divided into two parts – server side and client side implementations.
9-Aug-17	Iryna Sh.	3.61+ zh	Added new Balance Transfer API -> GetTransferStatus method.
10-Aug-17	Iryna Sh.	3.62	Added country and jurisdiction parameters to the Seamless Wallet API -> Authenticate response.
18-Aug-17	Iryna Sh.	3.63+ zh	Added new Data Feeds API -> Game Rounds method.
22-Aug-17	Iryna Sh.	3.64	Added new Integration API -> CancelRound method.
23-Aug-17	Iryna Sh.	3.65	Added new Data Feeds -> Daily Totals method.
31-Aug-17	Iryna Sh.	3.66	1. Added finished game rounds option to Data Feeds -> Game Rounds. 2. CSV format of Data Feeds -> Daily totals report has been replaced by JSON.

			3. Added new report Data Feeds -> Player's Incomplete Rounds.
6-Sep-17	Alex F.	3.67	Updated send Reconciliation retry call for Seamless Wallet API Error Codes
13-Nov-17	Alex F.	3.68	Added demo link for Asia Region
16-Nov-17	Iryna Sh.	3.69	Added optional jackpotContribution and jackpotId fields to Seamless Wallet API -> Bet. Configurable per brand.
7-Dec-17	Iryna Sh.	3.70	Free Rounds can be cancelled regardless of whether the player started to play them or not.
24-Jan-18	Iryna Sh.	3.71	Tournament API has been added.
30-Jan-18	Iryna Sh.	3.72	Date-time format was added to the Tournament API specs and examples.
27-Feb-18	Iryna Sh.	3.73	New betLimit parameter has been added to Seamless Wallet API -> Authenticate response, game Id is added to Authenticate request.
5-Mar-18	Iryna sh.	3.74	Error code 310 has been added to Seamless Wallet API.
23-Mar-18	Alex Ch.	3.75	OpenHistoryExtended has been added.
26-Apr-18	Alex Ch.	3.75	Data Feeds -> Daily Totals. Added "totalFreeSpinRounds" response parameter
4-May-18	Alex Ch.	3.75	Added: ipAddress to 3.3 Data Types, 3.4 Authenticate, 3.7 Bet, usedPromo to 3.3 Data Types
17-May-18	Iryna Sh.	3.76	Method Result: added parameters for Mystery Race winnings got by player during game round.
24-May-18	Iryna Sh.	3.77	Added Custom Messages chapter.
12-Jul-18	Iryna Sh.	3.78	Changes in Custom Messages: <ul style="list-style-type: none"> 1. All elements are optional, except message text 2. Added "nonIntrusive" Boolean parameter, to specify whether the messages should be a modal popup or a notification.
24-Jul-18	Iryna Sh.	3.79	Seamless Wallet API -> added new PromoWin method.
04-Oct-18	Sergiy B.	3.80	Seamless Wallet API - Added Language parameter to bet method. Renamed Mystery Race >>> Prize Drop
11-Nov-18	Alex Ch.	3.81	Added new section: "Responsible gaming – Sweden"
24-Dec-18	Iryna Sh.	3.82	Custom messages: new "open" link type was added for opening a new/blank tab in the browser, with the link specified for the action.
23-Jan-19	Iryna Sh.	3.83	Optional roundId and gameId parameters have been added to Seamless Wallet API -> bonusWin request, for compliance with the Spanish regulation.
31-Jan-19	Sergiy B	3.84	Adding section "XVI. Bingo Integration API"
04-Feb-19	Sergiy B	3.85	Adding optional parameter "language" to 7.7 openHistoryExtended
12-Mar-19	Sergiy B	3.86	Adding optional parameter "platform" to 8.1 Game Rounds
01-Apr-19	Sergiy B	3.87	Adding optional FRB-related parameters to 2.1 GetCasinoGames Updated section 1.1 Seamless Wallet API
26-Apr-19	Serhii B	3.88	Replacing parameters "date" and "hour" by "dateFrom" and "dateTo" in 16.4 AvailableGames Removing parameter "bingoType" from array Room in 16.4 AvailableGames Adding parameters "jurisdiction", "timeGameStart", "gamePrize", "jackpotAmount" to 16.4 AvailableGames Removing parameters "date" and "hour" 16.4 PreorderCards Adding parameters "timeGameStart" to 16.4 PreorderCards Replacing parameter "date" by "dateFrom" and "dateTo" in 16.4 JackpotWinners Changed behavior for mini-games in 16.1
10-May-19	Sergiy B	3.89	Added Section XVII Live Casino Dynamic Games Access
15-May-19	Iryna Sh.	3.90	Added new section Regulated markets including Italian Checksums API.
27-May-19	Iryna Sh.	3.91	Added two new type of Promo Win: C – Chat game win in bingo, B – Bonus award
12-June-19	Sergiy B.	3.92	Added "Nickname" method in 16.2 Bingo integration API for OPERATOR side
18-Jun-19	Iryna Sh.	3.93	Variable FRB API -> Create Free Rounds: <ul style="list-style-type: none"> • Added two optional parameters type and timeLimit for time limited FRB

			<ul style="list-style-type: none"> Parameter rounds gets optional.
09-July-19	Sergiy B.	3.94	Added optional parameter "lines" in 2.1 GetCasinoGames
12-July-19	Sergiy B.	3.95	Added section XIV Regulated Markets, 14.2 Portuguese Regulated Market
25-July-19	Sergiy B.	3.96	Added GG (Alderney) Jurisdiction
01-Aug-19	Sergiy B.	3.97	Added optional parameter "bonusCode" to 8.1, optional parameter "status" in 8.2
19-Sept-19	Sergiy B.	3.98	Removed "pull" method from XVII. Live Casino Dynamic Games Access
16-Oct_19	Sergiy B.	3.99	Added jurisdiction parameter to XV. Integration with demo games
17-Oct_19	Serhii B	3.100	XVI. Bingo Integration API: Changes in launching url parameters in order to support Sweden jurisdiction
17-Dec-19	Sergiy B.	3.101	3.12 Refund – refund of win transaction description removed
26-Dec-19	Sergiy B.	3.102	3.7 Bet - added mainbet/sidebet sending option in roundDetails
27-Dec-19	Sergiy B.	3.103	Added 3.16 SessionExpired
10-Jan-20	Sergiy B.	3.104	Added stylename parameter to XV. Integration with demo games
13-Jan-20	Sergiy B.	3.105	Section 17 DGA – Removed "photo" and "lang" parameters as unsupported
20-Jan-20	Sergiy B.	3.106	Section 17 DGA – Updated with new DGA version
11-Feb-20	Sergiy B.	3.107	3.16 SessionExpired – added optional token parameter in request
19-feb-20	Sergiy B.	3.108	Section 17 DGA – Updated with new DGA version 1.19.1
03-mar-20	Sergiy B.	3.109	Section 16.1 Bingo room Launching: deleted Bingolib.js, added direct URL explanation Section 16.3/ AvailableGames – changed type of gameId (number>>string) Section 3.3 DataTypes – added BY jurisdiction Added section 2.5 HealthCheck
02-apr-20	Sergiy B.	3.110	Required gameld parameter added to CancelRound, OpenHistoryExtended
10-apr-20	Sergiy B.	3.111	Optional dataType parameter added to VIII Data Feeds and Reports
14-apr-20	Sergiy B.	3.112	3.17 Error codes - added error code 130 for endRound
17-apr-20	Sergiy B.	3.113	3.4 Authenticate - added optional parameters extMinTotalBet and extMaxTotalBet
24-apr-20	Sergiy B.	3.114	Added section X. Business Statistics API +ZH
30-apr-20	Sergiy B.	3.115	Added jurisdiction, changed path in 15.2 Checksums API
06-may-20	Alex Ch	3.116	Added 17.4 Free Tickets API section. Added additional –methods in 17.3 Bingo integration API for PROVIDER side: Available sessions, SessionGameInfo, BuySession.
14-may-20	Sergiy B.	3.117	Added totalFreeSpinRounds in Business Statistics API, removed 17.5 Bingo Free rounds API, changed token data type to String(100)
09-Jun-20	Sergiy B.	3.118	Updated footings in I. API overview / 1.1 Seamless wallet API
10-Jun-20	Sergiy B.	3.119	15.3 Portuguese Slot Machine Reporting updated: sm_result >> smResult; XML>>JSON
20-Jun-20	Alex Ch	3.120	Updated 'Room opening' 'RoomList' 'ActiveGameConfigs' in terms of new 'Roll On' functionality. Fully reworked 'Error codes' sections.
26-Jun-20	Sergiy B.	3.121	XVI Integration with demogames – added description for stylename 3.7 Bet – added jackpotDetails ; 3.11 endRound – added win parameter
14-Jul -20	Sergiy B.	3.122	Added IM, RS jurisdictions
12-Aug -20	Sergiy B.	3.123	4.10 Error codes updated, retry policy added
14-Aug -20	Sergiy B.	3.124	III Seamless Wallet API - application/x-www-form-urlencoded; charset=UTF-8 added
19-Aug -20	Sergiy B.	3.125	added section 6.7 Create Player Free Rounds
01-Sep -20	Sergiy B.	3.126	added section XIX .Games JS API
30-Sep-20	Sergiy B.	3.127	Added Bingo FRBprizes (EndRound, Result, BonusWin), Bingo integration API for ROVIDER side; 3.1 Game opening - added explicit and GameLibJs launch options Added validityDate in 5.3 CreateFRB; Added 19.4 Live Casino Games Full Screen JS
09-Oct -20	Sergiy B.	3.128	Removed timeLimit Parameter and type "T" from Variable FRB / Create Free Rounds
16-Oct -20	Sergiy B.	3.129	Added dataType parameter in 2.1GetCasinoGames response

02-Nov -20	Sergiy B.	3.130	Added optional datatype for roundDetails in 3.7 Bet
10-Nov -20	Sergiy B.	3.131	Added 7.8 Error codes; added transfer retry important note to 4.5 Transfer

I. API Overview

Casino Game API supports different types of games: video slots, video poker, keno, etc. and API should not be changed if Casino Operator needs to add more games to their portfolio.

As soon as new game is available, it is added to Game Library with game symbols. Casino Operator will use provided symbol for StartGame method during opening specific game and Bet requests will be also sent using this symbol. Game Library will be provided to Casino Operator together with integration package or resent when new games are available.

Using this Casino Game API Casino Operator is able to integrate both types of games: web version for desktop devices and mobile version.

Platform provides the set of games built in different technologies – Flash and HTML5.

Information about what games have mobile version and available technologies for specific game is added to Game Library.

1.1 Seamless Wallet API

Operator should provide the Seamless Wallet Integration API on their side. Pragmatic Play will call the methods when players make a bet or get a win and their balance should be updated.

Method	Description	Status
Authenticate ¹	This method authenticates player by secure token. Casino Operator generates one-time secure token and sends it with the URL for opening Pragmatic Play game. See “ <i>Flow diagram of opening casino game</i> ”	Required
Balance	Returns the balance of a player.	Required
Bet	Checks if the player has enough funds and subtracts money from player’s balance. Returns the value of updated balance.	Required
Refund	Refunds player’s balance. This method will be used for cancellation of a bet in the case the game cannot be finished.	Required
Result	Adds winning amount to player’s balance. Returns the value of updated balance.	Required
BonusWin	Notifies the Casino Operator that free rounds are over and player’s balance should be increased with bonus amount.	Required
JackpotWin	Using this method a Pragmatic Play system will notify Casino Operator about Jackpot winning.	Required
PromoWin	Notifies the Casino Operator that tournament campaign is over and player’s <u>cash</u> balance should be increased with amount in promoWin.	Required
EndRound	Notifies the Casino Operator that the game round is over, for transactions finalization purpose.	Optional
SessionExpired	Notifies the Casino Operator that player’s game session has expired in PragmaticPlay system due to inactivity or game client closing.	Optional
Withdraw ²	Notifies the Casino Operator that a player clicked the <i>Withdraw</i> button in the lobby of downloadable client.	Optional
GetBalancePerGame	Returns the balance of a player per game.	Optional

¹ Authenticate2 for downloadable client

² for downloadable client only

1.2 Balance Transfer API

Operator can use this API for game opening and transferring funds to player's balance in the Pragmatic Play system.

Method	Description	Status
CreatePlayer	This method allows registering a new player on Pragmatic Play side.	Required
Transfer	This method transfers funds in to player's balance (i.e. deposit) and out of the player's balance (i.e. withdrawal) within Pragmatic Play system	Required
GetBalance	Using this method Casino Operator can get the current balance of the player in the Pragmatic Play system	Required
StartGame	Casino Operator should call this method before opening the game in player's browser. Secure one-time token is generated on the Pragmatic Play side. As the result of this method Casino Operator will get an URL to the requested game.	Required
TerminateSession	Using this method Casino Operator can terminate the current player session and kick him out all games	Required

1.3 Integration API

Integration API offers generic methods that allow Operators to create a game lobby, get statistics and close player sessions forcefully.

Method	Description	Status
GetCasinoGames	Using this method Casino Operator can retrieve the list of games available for integration. May be used for automatic building games lobby at casino website.	Optional
DownloadStatistics ^{3*}	This method may be used by Operator to store on the Pragmatic Play side the information about downloads and installations of the downloaded client.	Optional
CloseSession	This method terminates active game sessions of the player.	Optional

1.4 Data feeds

Reporting API on the Pragmatic Play side provides a set of data feeds for downloading play sessions, failed transactions and recent information about jackpots.

Method	Description	Status
Play Sessions	This data feed returns play sessions that was played during specified period.	Optional
Failed Transactions	Casino Operator can get the list of Refund and Result transactions that was not processed even after multiple attempts made by the Pragmatic Play server.	Optional
Active Jackpots	Provides information about open jackpots per casino brand.	Optional
Jackpot Winnings	Provides a list of last winnings of the specified jackpot.	Optional

³ for downloadable client only

1.5 Player ID

Seamless Wallet (Authenticate method)

Player ID (userId parameter) is unique identifier of the user within Casino Operator system. Before sending to Pragmatic Play any gaming related request Casino Operator should authenticate a player using Authenticate method. If player is new and its account does not exist in the Pragmatic Play system it will be created automatically on the base of the data sent by Casino Operator server in the Authenticate response. If player account already exists in the Pragmatic Play database it will be updated with the response data if necessary. Player id received in the Authenticate response will be sent with all subsequent requests to Casino Operator.

Balance Transfer (StartGame method)

Player ID is unique identifier of the user within Pragmatic Play system. Before sending to Pragmatic Play any gaming related request Casino Operator should register a player using CreatePlayer method, get player id and save it somewhere inside own system. Then this player id will be sent with all subsequent requests to both Casino Operator and Pragmatic Play.

Please note: this type of game opening remains in the API for compatibility with the previous versions of the interface. Option 2 is more standard and preferred way to open games for integrated systems.

1.6 Play session

Play session is a game round in which bet and wins are combined together. Each round can contain several bets, win and refunds of the bets.

1.7 Transaction reference

Transaction reference is unique transaction id within Pragmatic Play system. Transaction reference used for bets and wins has to be different.

II. Integration API

This is a simple API for Casino Operator to retrieve some data from Pragmatic Play system and send some events to Pragmatic Play. API is an HTTP listener, which listens POST requests coming to an URL with the request mappings below.

All responses are in JSON format.

URL of the HTTP service will be provided by Pragmatic Play for the production and test environments and looks like:

`https://{API service domain}/IntegrationService/v3/http/CasinoGameAPI`

Integration simple HTTP Service is securely protected, hence please be sure:

- player's browser (end) must NOT be used as initiator of requests to the service;
- proper IPs are supplied to Pragmatic Play for whitelisting

2.1 GetCasinoGames

Request path: POST /getCasinoGames/

Using this method Casino Operator will get the list of casino games available for the integration.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
options	List of settings (String). By including it, operator can get additional information about game. Possible values are: <ul style="list-style-type: none">• GetFrbDetails• GetLines• GetDataTypes	Optional
hash	Hash code of the request.	Required

Response parameters

Name	Type	Description
gameList	CasinoGame	The list of CasinoGame objects for transferring casino games details (see data type description below)
error	String(10)	0 - if the request was successfully processed or error code in other case.
description	String(250)	Description of the error for troubleshooting purposes.

Type: CasinoGame

Name	Type	Description	Status
gameID	String(20)	Symbolic unique identifier of the game provided by Pragmatic Play. <i>Example: vs7monkeys, vs50amt, rla, wra, bjb</i>	Required
gameName	String(100)	Name of the game. <i>Example: "Lucky Dragons", "Hot Safari"</i>	Required

gameTypeID	String(20)	Symbolic unique identifier of the game. <i>Example: "vs", "rl", "vp".</i>	Required
typeDescription	String(100)	Short description of the game type. <i>Example: "Video Slot", "Roulette", "Video Poker".</i>	Required
technology	String(30)	Short description of the technologies in which the game is available, comma separated. Possible values are: <ul style="list-style-type: none"> html5 - HTML5 games flash – Flash (only for desktop) <i>Example: "flash,html5", "html5"</i>	Required
technologyID	String(10)	Technologies in which the game is available, comma separated. Possible values are: <ul style="list-style-type: none"> H5 - HTML5 games F – Flash (only for desktop) <i>Example: "H5", "F,H5"</i>	Required
platform	String(30)	Platform for which the game can be opened. Possible values, comma separated: <ul style="list-style-type: none"> MOBILE – if game should be opened on mobile devices WEB – if game will be opened on desktop devices DOWNLOAD – if game can be opened in downloaded gaming client 	Required
demoGameAvailable	Boolean	If true, a demo version of the game is available.	Required
aspectRatio	String(10)	Describes the proportional relationship between the width and the height of the game. <i>Example: "4:3", "16:9"</i>	Required
gameIDNumeric	Long	The numeric value of gameID	Required
frbAvailable	Boolean	If true, a free round bonus of the game is available. <i>This field is optional and will appear in response in cases where request contains GetFrbDetails in options list</i>	Optional
variableFrbAvailable	Boolean	If true, a variable free round bonus of the game is available <i>This field is optional and will appear in response in cases where request contains GetFrbDetails in options list</i>	Optional
lines	Integer	Number of available pay lines in game <i>This field is optional and will appear in response in cases where request contains GetLines in options list</i>	Optional
dataType	String(3)	Type of game portfolio Available options: RNG - Main portfolio games (video slots, classic slots etc.) LC - Live Casino portfolio R2 - Fishing games portfolio VSB - Virtual sports betting portfolio <i>This field is optional and will appear in response in cases where request contains GetDataTypes in options list</i>	Optional

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/getCasinoGames/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&hash=da77af8d0ba97eaa033682466b87f569&options=GetFrbDetails,GetLines,
GetDataTypes
```

Example of JSON response:

```
{
  "description": "OK",
  "error": "0",
  "gameList": [
    {
      "gameID": "vs7776secrets",
      "gameName": "Aztec Treasure",
      "gameTypeID": "vs",
      "typeDescription": "Video Slots",
      "technology": "html5",
      "platform": "MOBILE,DOWNLOAD,WEB",
      "demoGameAvailable": true,
      "aspectRatio": "16:9",
      "technologyID": "H5",
      "gameIdNumeric": 1550843208,
      "jurisdictions": [
        "99"
      ],
      "frbAvailable": true,
      "variableFrbAvailable": true,
      "lines": 20
      "dataType": "RNG"
    },
    ...,
    {
      "gameID": "vs243mwarrior",
      "gameName": "Monkey Warrior",
      "gameTypeID": "vs",
      "typeDescription": "Video Slots",
      "technology": "html5",
      "platform": "MOBILE,DOWNLOAD,WEB",
      "demoGameAvailable": true,
      "aspectRatio": "16:9",
      "technologyID": "H5",
      "gameIdNumeric": 1553615521,
      "jurisdictions": [
        "99",
        "BG",
        "CO",
        "DK",
        "PT",
        "EE",
        "IT",
        "LV",
        "MT",
        "LT",
        "SE",
        "UK",
        "RO",
        "ES"
      ],
      "frbAvailable": true,
      "variableFrbAvailable": true,
      "lines": 25
      "dataType": "RNG"
    }
  ]
}
```


2.1.1 Game icons

Operator can use URLs below in order to get icons for the games:

Rectangle, size 325x234:

`http(s)://{game server domain}/game_pic/rec/325/{gameID}.png`

Rectangle, size 188x83:

`http(s)://{game server domain}/game_pic/rec/188/{gameID}.png`

Rectangle, size 160x115:

`http(s)://{game server domain}/game_pic/rec/160/{gameID}.png`

Square, size 200x200:

`http(s)://{game server domain}/game_pic/square/200/{gameID}.png`

Square, size 138x138:

`http(s)://{game server domain}/game_pic/square/138/{gameID}.jpg`

Example of HTTP request:

```
GET /game_pic/rec/325/vs50aladdin.png HTTP/1.1
Host: {game server domain}
Cache-Control: no-cache
```

Example of response:



2.2 DownloadStatistics

Request path: POST /downloadStatistics/

This method can be used by Operator in case they need to gather events related to the downloaded client. All events sent by Operator are stored in the database and then aggregated information about number of downloads and client installations is available in the back office application.

Request parameters

Name	Description
secureLogin	User name for authentication in the Casino Game API service
playerId	Id of the player within the Operator system.
referrerUrl	The referrer URL.
ipAddress	IP address of the player.
clientOS	Type of operational system of the player's device. Available values are: A - Android I - iOS M - Mac U - Unix W - Windows O - Others
eventType	Type of the event. Available values are: D – Download started S – Installation started C – Installation completed.
hash	Hash code of the request.

To get reliable download statistics Operator should send only unique download events. This may be achieved by using cookies on Operator's site.

For example, when player downloads the client for the first time, a cookie can be saved on the player's computer. Every time player clicks "Download" button the site should check the "download cookie" and send download event only if the cookie is not found.

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/downloadStatistics/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&playerId=421&referrerUrl=domain.com&ipAddress=1.1.1.1&clientOS=W&eventType=D&hash=5fd31717ba4da9e78ac19730212925c2
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK"
}
```

2.3 CloseSession

Request path: POST /closeSession/

This method terminates active game sessions of the player. Operator has the option to remove player's history, so that the incomplete game rounds cannot be finished by another player (usually this is relevant for terminals). To close only game session for specific game Operator may send game id parameter in the request.

Request parameters

Name	Description
secureLogin	User name for authentication in the Casino Game API service
externalPlayerId	Id of the player within the Operator system.
gameId	Id of the game. This is optional parameter, which has to be sent by Operator if only the session for specific game should be closed.
clearHistory	Specifies whether to clear the history of the round or not. Optional parameter, default value is 0. May have the following values: 1 – history should be removed, so that the last game round cannot be completed anymore 0 – last game round can be completed
hash	Hash code of the request.

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/closeSession/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&externalPlayerId=421&gameId=vs7monkeys&clearHistory=1&hash=5fd31717ba4da9e78ac19730212925c2
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK"
}
```

2.4 CancelRound

Request path: POST /cancelRound/

CancelRound method initiates a refund request, which the Pragmatic system will send to the Operator's remote wallet. After successful refund call the game round will be marked as Cancelled in the Pragmatic system.

This method Operator can use any time they want to close player's round forcefully, due to a retention policy on the Operator's system or according to requirements for regulated markets.

Request parameters

Name	Description
secureLogin	User name for authentication in the Casino Game API service
externalPlayerId	Id of the player within the Operator system.
gameId	Id of the game. This is required parameter.
roundId	Id of the game round to be canceled (play session id).
hash	Hash code of the request.

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/cancelRound/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&externalPlayerId=421&gameId=vs7monkeys&roundId=123456&hash=5fd31717ba4da9e78ac19730212925c2
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK"
}
```

2.5 HealthCheck

HealthCheck method can be used by operators to check if PragmaticPlay provider API service or Game server are live and ready.

API service HealthCheck

Request path:

GET https://{API-service-domain}/IntegrationService/v3/http/CasinoGameAPI/health/heartbeatCheck

Game server HealthCheck

Request path:

http://{game-server-domain}/gs2c/livetest

Frequency of calling HealthCheck should not be more than 1 call per minute for any particular brand.

Response statuses

HTTP Status	Description
200	API service is available to use
Not 200	Any other HTTP status means some issues with API service. Not ready to use.

Example for API Service HealthCheck

Example of HTTP request:

```
GET /IntegrationService/v3/http/CasinoGameAPI/health/heartbeatCheck HTTP/1.1
Host: api.prerelease-env.biz
```

Example of JSON response:

```
HTTP Status: 200 OK

Response body:
{
  "error": "0",
  "description": "OK"
}
```

Example for Game server HealthCheck

Example of HTTP request:

```
GET /gs2c/livetest HTTP/1.1
Host: ext-test1.prerelease-env.biz
```

Example of response:

```
HTTP Status: 200 OK
```

III. Seamless Wallet API

This is a simple API for Pragmatic Play gaming platform to connect to player's wallet. API is an HTTP listener, which listens application/x-www-form-urlencoded POST requests.

In POST requests can be sent either HTTP Header:

- Content-Type: application/x-www-form-urlencoded
- Content-Type: application/x-www-form-urlencoded; charset=UTF-8

For application/x-www-form-urlencoded, the body of the HTTP message sent to the Seamless Wallet API is essentially one giant query string where name/value pairs are separated by the ampersand (&), and names are separated from values by the equals symbol (=). An example of this would be (please see request examples for each call below):

parameter1=value1¶meter2=value2

All responses should be in JSON format (please see response examples for each call below).

URL of the Seamless Wallet API should be provided by Casino Operator for the production and test environments.

3.1 Game opening

There are 2 possible ways to compose launch URL. Operator can form URL explicitly or call JS function, to get launch URL:

Explicit launch URL

In cases where operator composes launch URL, it should contain following parameters:

- **game server domain** – domain name for game opening and playing;
- **key** – URL-encoded list of parameters:
 - **token** – secure one-time token is generated by OPERATOR for specific player;
 - **symbol** – unique identifier for the game, e.g. *vs25queenofgold*, *vs50aladdin*, *cs5triple8gold*, *bjma*, *vpa*, etc.;
 - **language** – player language in ISO 639-1 standard (e.g. en, fr, it);
 - **technology** – *H5* or *F* (for *WEB platform*) or *H5* (for *MOBILE platform*);
 - **platform** – *WEB* (for *desktop devices*) or *MOBILE* (for *mobile devices*)
 - **cashierUrl** – an URL for opening the Operator's website Cashier page;
 - **lobbyUrl** – an URL for opening the Operator's website Lobby page;
- **stylename** – unique identifier of the OPERATOR at PROVIDER side.
- **rci** – the reality check interval, in minutes. . This is optional parameter;
- **rce** – the reality check elapsed time, in minutes. This is optional parameter;
- **operatorGameHistoryUrl** – an URL to open game history page on Operator's side
- **rcHistoryUrl** – a link to the player's game history. This is optional parameter;
- **rcCloseUrl** – a link to the page on the Operator's website, to which the player will be redirected if they choose to close the game. This parameter is optional;

Example of game opening link:

```
https://{game_server_domain}/playGame.do?key=token%3Drgklr4d5b6%26symbol%3Dvs20bl%26technology%3DH5%26platform%3D%26language%3Den%26cashierUrl%3Dhttp%3A%2F%2Fvs20bl%26lobbyUrl%3D&stylename=username
```

GameLib JS function

To get an URL for game opening Operator may call JS function `GameLib.gameUrl` (which is available via `GameLib.js` at `https://{game server domain}/gs2c/common/js/lobby/GameLib.js`) with the following parameters:

game server domain – a domain name for game opening and playing

token – secure token generated by operator for the player
symbol – unique identifier for the game, e.g. *vs25queenofgold*, *vs50aladdin*, *cs5triple8gold*, *bjma*, *rla*, *vpa*, etc.
technology – H5 or F (for WEB platform) or H5 (for MOBILE platform)
platform – WEB (for desktop devices) or MOBILE (for mobile devices)
language – player language in ISO 639-1 standard (e.g. en, fr, it)
cashierUrl – an URL for opening the Operator’s website Cashier page;
lobbyUrl – an URL for opening the Operator’s website Lobby page;
operatorGameHistoryUrl – an URL to open game history page on Operator’s side
secureLogin – unique identifier of the Operator at Pragmatic Play side.

So, the function call should look like:

```
var launchurl = GameLib.gameUrl(domain, token, symbol, technology, platform, language, cashierUrl, lobbyUrl, secureLogin)
```

As a result of function call Operator gets an URL for game opening. The link will look like:

```
https://{game server domain}/gs2c/playGame.do?key=token%3D{token}%26symbol%3D{symbol}%26technology%3D{technology}%26platform%3D{platform}%26language%3D{language}%26cashierUrl%3DcashierUrl%26lobbyUrl%3DlobbyUrl&styleName={secureLogin}
```

where all values in {...} are equal to GameLib.gameUrl function call parameters.

3.2 Hash calculation

Hash code is calculated with following formula:

From request post parameters, all parameters are taken (except hash) and appending to string:

1. Sort all parameter by keys in alphabetical order.
2. Append them (if the value is not null or empty) in key1=value1&key2=value2.
3. Append secret key, e.g.: key1=value1&key2=value2SECRET.
4. Calculate the hash by using MD5.
5. Compare with hash parameter. In the case of failure Casino Operator should send the error code 5.

3.3 Data types

Parameter	Type	Description
userId	String(100)	Unique identifier of the user within Casino Operator system. Parameter value is case-sensitive. <i>Example: joe1001, 2644987, playerABC*, playerAbc*</i> * playerABC and playerAbc – are two different player accounts within Pragmatic Play system
currency	String(3)	Player's ISO 4217 currency code. <i>Example: EUR, USD</i>
country	String(2)	Country of the player. 2-letter Country code, ISO 3166-1 alpha-2. <i>Examples: GB, GI, DK, ES, IT, PT</i>
jurisdiction	String(2)	Jurisdiction of the player. Available values are: <ul style="list-style-type: none">○ BS – Bahamas○ BG – Bulgaria○ DK – Denmark○ UK – United Kingdom○ IT – Italy○ ES – Spain○ PT – Portugal○ RO – Romania○ MT – Malta○ PH – Philippines○ LT – Lithuania○ LV – Latvia○ EE – Estonia○ SE – Sweden○ CO – Colombia○ GG – Guernsey (Alderney)○ BY – Belarus○ RS – Serbia○ IM – Isle of Man○ GR – Greece○ DE – Germany○ 99 – Unregulated market
cash	Decimal (10, 2)	<i>Player's real balance.</i> <i>Example: 100.00</i>
bonus	Decimal (10, 2)	<i>Player's promo balance.</i> <i>Example: 100.00</i>
amount	Decimal (10, 2)	<i>Amount of the bet or win.</i> <i>Example: 1.00</i>
win	Decimal (10, 2)	<i>Amount of win, which can be optionally sent in endRound calls.</i> <i>Example: 1.00, 0.0</i>
roundId	Long	Identification for the player's round.
jackpotId	Long	Identification for the jackpot.
jackpotContribution	Decimal (10, 6)	Amount of the contribution to the jackpot
jackpotDetails	String (Valid JSON)	Detailed contribution amounts for multi-tier jackpots

gameId	String (32)	Identifier of the game. <i>Example: vs25queenofgold, vs50aladdin, cs5triple8gold, bjma, rla, vpa</i> The list of available games may be obtained by Casino Operator by calling GetCasinoGames method of CasinoGameAPI of Integration simple HTTP Service.
language	String(2)	Language code in ISO 639-1 standard (e.g. en, fr, it).
reference	String (32)	Unique reference of the transaction.
sessionId	String (100)	Game session ID on Pragmatic Play side
token	String (100)	Authentication token of the player generated by the Casino Operator system. Token is passed to Pragmatic Play as a game launch request parameter or as a authenticate response parameter afterwards.
providerId	String (32)	Game provider identifier. May be provided by Casino Operator. Otherwise default is used.
transactionId	Varchar (32)	Unique ID of the transaction within Casino Operator system.
timestamp	Long	Date and time when the transaction is processed on the Pragmatic Play side (<i>Unix epoch time in milliseconds, for example : 1470926696715</i>)
roundDetails	String(50)	Additional information about the current game round, such as “gamble”, “spin”, “bonus” - comma separated. Examples: <ul style="list-style-type: none"> • “spin” • “type:main,desc:Tie” • “type:betbehind,desc:Bet behind Play seat 7,”
bonusCode	String (100)	Unique id of the Free Round Bonus within Casino Operator system.
platform	String(10)	The platform type (channel) on which the game is played. Possible values: <ul style="list-style-type: none"> • “MOBILE” – mobile device • “WEB” – desktop device • “DOWNLOAD” – downloadable client
error	Integer	Error code.
usedPromo	Decimal (10, 2)	Part of bet amount which is taken from bonus balance
ipAddress	String (32)	IP address of the player
campaignId	Long	Id of the campaign
campaignType	String(1)	Type of the campaign
promoWinAmount	Decimal (10, 2)	Prize amount that the player is awarded with during a promotional campaign
promoWinReference	String (32)	Unique reference of the promotional transaction.
promoCampaignID	Long	Id of the promotional campaign.
promoCampaignType	String(1)	Type of the promotional campaign.
specPrizeAmount	Integer	number of FRB which was won by player in bingo game
specPrizeCode	String (100)	unique FRB bonus code within PP system
specPrizeType	String(3)	Type of free special prize Possible values: <ul style="list-style-type: none"> • “FRB” – free round bonuses

3.4 Authenticate

Request path: POST /authenticate.html

When the game is opening Pragmatic Play receives with URL security token generated by Casino Operator. Using this token Pragmatic Play will ask Casino Operator for player authentication and get the player's balance.

Important: Token from Authenticate response can be used in subsequent wallet requests for other games, launched from the inside of game or bingo client.

Pragmatic Play can send subsequent authenticate requests with the same token, when player:

- opens new casino game from built-in mini lobby
- opens bingo room from bingo lobby
- opens live casino game from live casino lobby

Sending additional Authenticate requests for inner opening is optional *and is not handled by the Pragmatic Play by default. In case Casino Operator need to receive additional authorization for inner opening - they should ask the Pragmatic Play's Technical Support for configuration.*

Operator should accept additional authenticate requests with active token for player.

Request parameters

Name	Description	Status
hash	Hash code of the request.	Required
token	Token of the player.	Required
providerId	Game Provider identifier.	Required
gameId	Id of the game. <i>The field is optional and is not handled by the Pragmatic Play by default. In case Casino Operator need individual bet limits to be applied when the game opens they should ask the Pragmatic Play's Technical Support for configuration.</i>	Optional
ipAddress	IP address of the player. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional

Response parameters

Name	Description	Status
userId	Identifier of the user within the Casino Operator's system.	Required
currency	Currency of the player.	Required
cash	Real balance of the player.	Required
bonus	Bonus balance of the player.	Required
token	Token/session of the player. Can be configured to be returned in all other API calls so Operator has control over it across playing process. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional

country	Country of the player.	Optional
jurisdiction	Jurisdiction of the player. Please see the <i>Data types</i> section for more details.	Optional
betLimits	<p>An optional element. How this is used is supplier specific, for example it might constitute a further restriction on limits already in place on the game. Structure of the element:</p> <p>defaultBet – the default bet value</p> <p>minBet – the minimal individual bet amount</p> <p>maxBet – the maximal individual bet amount</p> <p>maxTotalBet – the maximum total bet amount (in the game currency)</p> <p>minTotalBet – the minimum total bet amount (in the game currency)</p> <p><i>optionally for in-game buy feature:</i></p> <p>extMinTotalBet – the minimum total bet amount for in-game buy feature (in the game currency)</p> <p>extMaxTotalBet – the maximum total bet amount for in-game buy feature (in the game currency)</p> <p><i>If extMinTotalBet and extMaxTotalBet are not included in response, default values for in-game buy feature will be used</i></p> <p><i>The field is optional and is not handled by the Pragmatic Play by default. In case Casino Operator need individual bet limits to be applied when the game opens they should ask the Pragmatic Play's Technical Support for configuration.</i></p>	Optional

The currency of the player is set up when the player opens a game for the first time, and the player account is created in the Pragmatic Play database, with the userId and currency received in the response on the Authentication request.

Important: Currency cannot be changed after the player account is created.

Example of HTTP request:

```
POST /authenticate.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

providerId=pragmaticplay&hash=e1467eb30743fb0a180ed141a26c58f7&token=5v93mto7jr
```

Example of JSON response:

```
{
  "userId": "421",
  "currency": "USD",
  "cash": 99999.99,
  "bonus": 99.99,
  "country": "GB",
  "jurisdiction": "UK",
  "betLimits":
  {
    "defaultBet": 0.10,
    "minBet": 0.02,
    "maxBet": 10.00,
    "minTotalBet": 0.50,
    "maxTotalBet": 250.00,
  }
  "error": 0,
  "description": "Success"
}
```

3.5 Authenticate2

Request path: POST /authenticate2.html

Authenticate2 method should be implemented only in the case when Operator has requested the downloadable client.

When a player logs into the client Pragmatic Play receives the username and password provided by the player. Using these credentials Pragmatic Play will ask Casino Operator's system for the player authentication and get the player's balance.

Request parameters

Name	Description	Status
hash	Hash code of the request.	Required
username	Username of the player.	Required
password	Password provided by the player.	Required
providerId	Game provider identifier.	Required

Response parameters

Name	Description	Status
userId	Identifier of the user within the Casino Operator's system.	Required
currency	Currency of the player.	Required
cash	Real balance of the player.	Required
bonus	Bonus balance of the player.	Required

The currency of the player is set up when the player opens a game for the first time, and the player account is created in the Pragmatic Play database, with the userId and currency received in the response on the Authentication request.

Important: Currency cannot be changed after the player account is created.

Example of HTTP request:

```
POST /authenticate2.html HTTP/1.1
Host: someoperatoromain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

providerId=pragmaticplay&username=testuser&password=testpassword&hash=ccc7e0436f56b9d04c0a5fed92c05be9
```

Example of JSON response:

```
{
  "userId": "421",
  "currency": "USD",
  "cash": 99999.99,
  "bonus": 99.99,
  "error": 0,
  "description": "Success",
}
```

3.6 Balance

Request path: POST /balance.html

Using this method a Pragmatic Play system will know a current balance of player and will show it in the game.

Request parameters

Name	Description	Status
hash	Hash code of the request.	Required
providerId	Game Provider identifier.	Required
userId	Identifier of the user within the Casino Operator's system.	Required
token	Token of the player from Authenticate response. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional

Response parameters

Name	Description	Status
currency	Currency of the player.	Required
cash	Real balance of the player.	Required
bonus	Bonus balance of the player.	Required

Example of HTTP request:

```
POST /balance.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

providerId=pragmaticplay&userId=421&hash=b4672931ee1d78e4022faaadf58e37db
```

Example of JSON response:

```
{
  "currency": "USD",
  "cash": 99999.99,
  "bonus": 99.99,
  "error": 0,
  "description": "Success"
}
```

3.7 Bet

Request path: `POST /bet.html`

Using this method Pragmatic Play system will check the player balance on Casino Operator side to ensure they still have the funds to cover the bet. Amount of the bet must be subtracted from player balance in Casino Operator system.

Important: The call is idempotent, i.e. sending bet again only creates one transaction.

Request parameters

Name	Description	Status
hash	Hash code of the request	Required
userId	Identifier of the user within the Casino Operator's system.	Required
gameId	Id of the game.	Required
roundId	Id of the round.	Required
amount	Amount of the bet. Minimum is 0.00.	Required
reference	Unique reference of this transaction.	Required
providerId	Game Provider id.	Required
timestamp	Date and time when the transaction is processed on the Pragmatic Play side (Unix epoch time in milliseconds, for example : 1470926696715)	Required
roundDetails	Additional information about the current game round. In case Casino Operator needs additional bet description for Live Casino Games to be sent (such as bet type and seat number), they should ask the Pragmatic Play's Technical Support for additional configuration. In this case Operator should accept data type: String(300) for roundDetails parameter in current method, Game History API , Data feeds and reports API . Examples of roundDetails parameter with enabled bet description: <ul style="list-style-type: none">"type:main,desc:Play seat 4,Play seat 3,Perfect pairs seat 4,Perfect pairs seat 3,Blackjack 21 + 3 seat 4,Blackjack 21 + 3 seat 3""type:betbehind,desc:Bet behind Play seat 7,""type:main,desc:Double down seat 4,"	Required
bonusCode	Id of the bonus in Casino Operator system. (*is mandatory in case of FRB API is implemented) The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.	Optional*
platform	The platform type (channel) on which the game is played. Possible values: <ul style="list-style-type: none">"MOBILE" – mobile device"WEB" – desktop device"DOWNLOAD" – downloadable client The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.	Optional
language	Language on which the game was opened. The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.	Optional
jackpotContribution	Amount of the contribution to the jackpot. If there is a multi-tier jackpot, this field will contain the total amount of contributions to all jackpots. The field is optional and should be sent together with jackpotId. The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.	Optional

jackpotDetails	Amounts of the contribution for multi-tier jackpot, separated by tiers. The field is optional and should be sent together with jackpotId and jackpotContribution. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional
jackpotId	Id of the active jackpot to contribute. The field is optional and should be sent together with jackpotContribution. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional
token	Token of the player from Authenticate response. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional
ipAddress	IP address of the player. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional

Response parameters

Name	Description	Status
transactionId	Id of the transaction in wallet.	Required
currency	Currency of the player.	Required
cash	Real balance of the player.	Required
bonus	Bonus balance of the player.	Required
usedPromo	Amount was used from the bonus balance.	Required

Example of HTTP request:

```
POST /bet.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

roundDetails=spin&reference=585c1306f89c56f5ecfc2f5d&gameId=vs50aladdin&amount=100.0&providerId=pragmaticplay&userId=421&roundId=5103188801&hash=4a5d375ac1311b04fba2ea66d067b8e5&timestamp=1482429190374
```

Example of JSON response:

```
{
  "transactionId": 1482429190474,
  "currency": "USD",
  "cash": 99899.99,
  "bonus": 99.99,
  "usedPromo": 0,
  "error": 0,
  "description": "Success"
}
```


3.8 Result

Request path: POST /result.html

Using this method the Pragmatic Play system will send to Casino Operator the winning result of a bet. The Casino Operator will change the balance of the player in accordance with this request and return the updated balance.

Result request may contain a prize that the player is awarded with during the game round, if there is an active promotional campaigns like Prize Drop. Parameters related to the Prize Drop prizes are optional and should be configured by PragmaticPlay team based on Operator's request.

Important: The call is idempotent, i.e. sending result again with the same reference number creates only one transaction.

Request parameters

Name	Description	Status
hash	Hash code of the request	Required
userId	Identifier of the user within the Casino Operator's system.	Required
gameId	Id of the game.	Required
roundId	Id of the round.	Required
amount	Amount of the win.	Required
reference	Unique reference of this transaction.	Required
providerId	Game Provider id.	Required
timestamp	Date and time when the transaction is processed on the Pragmatic Play side (Unix epoch time in milliseconds, for example : 1470926696715)	Required
roundDetails	Additional information about the current game round.	Required
bonusCode	Id of the bonus in Casino Operator system. (*is mandatory in case of FRB API is implemented) The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.	Optional*
platform	The platform type (channel) on which the game is played. Possible values: <ul style="list-style-type: none">• "MOBILE" – mobile device• "WEB" – desktop device• "DOWNLOAD" – downloadable client The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.	Optional
token	Token of the player from Authenticate response. The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.	Optional
promoWinAmount	Prize amount that the player is awarded with during a promotional campaign. The field is optional and will appear only if there is a winning in a campaign. Please note, that promoWinAmount, promoWinReference, promoCampaignID and promoCampaignType parameters should appear together.	Optional
promoWinReference	Unique reference of this transaction. The field is optional and will appear only if there is a winning in a campaign. Please note, that promoWinAmount, promoWinReference, promoCampaignID and promoCampaignType parameters should appear together.	Optional

promoCampaignID	Id of the promotional campaign. <i>The field is optional and will appear only if there is a winning in a campaign. Please note, that promoWinAmount, promoWinReference, promoCampaignID and promoCampaignType parameters should appear together.</i>	Optional
promoCampaignType	Type of the promotional campaign. Available values are: R – Prize Drop (Cash drop promotion) <i>The field is optional and will appear only if there is a winning in a campaign. Please note, that promoWinAmount, promoWinReference, promoCampaignID and promoCampaignType parameters should appear together.</i>	Optional
specPrizes[#].specPrizeAmount	number of FRB which was won by player in bingo game (where # - serial number of prize) <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional (Bingo only)
specPrizes[#].specPrizeCode	unique FRB bonus code within PP system (where # - serial number of prize) <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional (Bingo only)
specPrizes[#].specPrizeType	Type of free special prize (where # - serial number of prize) Possible values: <ul style="list-style-type: none"> “FRB” – free round bonuses <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional (Bingo only)

Response parameters

Name	Description	Status
transactionId	Id of the transaction in wallet.	Required
currency	Currency of the player.	Required
cash	Real balance of the player.	Required
bonus	Bonus balance of the player.	Required

Example of HTTP request:

```
POST /result.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

roundDetails=spin&reference=585c156df89c56f5ecfd99fb&gameId=vs50aladdin&amount=10.0&providerId=pragmaticplay&userId=421&roundId=5103268693&platform=DOWNLOAD&hash=533c609c6a74b533efb870b806f00732&timestamp=1482429805138
```

Example of JSON response:

```
{
  "transactionId": 1482429805253,
  "currency": "USD",
  "cash": 99809.99,
  "bonus": 99.99,
  "error": 0,
  "description": "Success"
}
```

3.9 BonusWin

Request path: *POST /bonusWin.html*

Using this method a Pragmatic Play system will send to Casino Operator winning result of all rounds played on Free Round Bonus. Casino Operator will change a player balance in appliance with this request and will return an updated balance.

Important: The call is idempotent, i.e. sending result again with the same reference number creates only one transaction.

Request parameters

Name	Description	Status
hash	Hash code of the request	Required
userId	Identifier of the user within the Casino Operator's system.	Required
amount	Amount of the win. Minimum is 0.00 (Zero amount is treated as loss).	Required
reference	Unique reference of this transaction.	Required
providerId	Game Provider id.	Required
timestamp	Date and time when the transaction is processed on the Pragmatic Play side (Unix epoch time in milliseconds, for example : 1470926696715)	Required
bonusCode	Id of the bonus in Casino Operator system. (*is mandatory in case of FRB API is implemented) <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional*
roundId	Id of the last played round in Free Round Bonus <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional
gameId	Id of the game for the last played round within Free Round Bonus <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional
token	Token of the player from Authenticate response. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional
specPrizes[#].specPrizeAmount	number of FRB which was won by player in bingo game (where # - serial number of prize) <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional (Bingo only)
specPrizes[#].specPrizeCode	unique FRB bonus code within PP system (where # - serial number of prize) <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional (Bingo only)
specPrizes[#].specPrizeType	Type of free special prize(where # - serial number of prize) Possible values: <ul style="list-style-type: none">• "FRB" – free round bonuses <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional (Bingo only)

Response parameters

Name	Description	Status
transactionId	Id of the transaction in wallet.	Required
currency	Currency of the player.	Required
cash	Real balance of the player.	Required
bonus	Bonus balance of the player.	Required

Example of HTTP request:

```
POST /bonusWin.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

reference=585d0257f89c56f5ed6b2e37&bonusCode=test_pp_frb1&amount=1.0&providerId=pragmaticplay&userId=421&hash=242d4d029c20e6e4692a4b88398f4fdc&timestamp=1482490455354
```

Example of JSON response:

```
{
  "transactionId": 1482490793341,
  "currency": "USD",
  "cash": 99710.04,
  "bonus": 99.99,
  "error": 0,
  "description": "Success"
}
```

3.10 JackpotWin

Request path: POST /jackpotWin.html

Using this method a Pragmatic Play system will notify Casino Operator about Jackpot winning. Notification is asynchronous and may come to the operator with a short delay after game round is over. Operator should handle the transaction in their system and send the jackpot win transaction id back to the Pragmatic Play.

Important: The call is idempotent, i.e. sending result again with the same reference number creates only one transaction.

Request parameters

Name	Description	Status
hash	Hash code of the request	Required
providerId	Game Provider id.	Required
timestamp	Date and time when the transaction is processed on the Pragmatic Play side (Unix epoch time in milliseconds, for example : 1470926696715)	Required
userId	Identifier of the user within the Casino Operator's system.	Required
gameId	Id of the game.	Required
roundId	Id of the game round.	Required
jackpotId	Id of the jackpot.	Required
amount	Amount of the jackpot winning.	Required
reference	Unique reference of the transaction within the Pragmatic Play system.	Required
platform	The platform type (channel) on which the game is played. Possible values: <ul style="list-style-type: none">• "MOBILE" – mobile device• "WEB" – desktop device• "DOWNLOAD" – downloadable client <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional
token	Token of the player from Authenticate response. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional

Response parameters

Name	Description	Status
transactionId	Id of the transaction in wallet.	Required
currency	Currency of the player.	Required
cash	Real balance of the player.	Required
bonus	Bonus balance of the player.	Required

Example of HTTP request:

```
POST /jackpotWin.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

reference=585d0b2af89c56f5ed6f0dlf&gameId=vs30catz_jp&amount=55.0&jackpotId=568&providerId=pragmaticplay&userId=421&roundId=5109164607&hash=8ef28798d5b3e523528bdb61ada939a7&timestamp=1482492714431
```

Example of JSON response:

```
{
  "transactionId": 1482492905503,
  "currency": "USD",
  "cash": 99815.04,
  "bonus": 99.99,
  "error": 0,
  "description": "Success"
}
```

3.11 EndRound

Request path: *POST /endRound.html*

Every time a game round is over, the Pragmatic Play system will call EndRound method, so that Operator can finalize the game round transactions on their side in real time.

If Operator does not need to finalize transactions in real time it is strongly recommended to disable EndRound functionality on the Pragmatic Play side. Instead, Data Feed API can be used for transaction finalization, in order to decrease the number of requests to the Operator system and keep traffic between two systems as light as possible.

Important: EndRound request may be sent more than once. If the game round is already finalized, Operator should ignore the EndRound request and return the success response.

Request parameters

Name	Description	Status
hash	Hash code of the request	Required
userId	Identifier of the user within the Casino Operator's system.	Required
gameId	Id of the game.	Required
roundId	Id of the round.	Required
providerId	Game Provider id.	Required
platform	<p>The platform type (channel) on which the game is played.</p> <p>Possible values:</p> <ul style="list-style-type: none">• "MOBILE" – mobile device• "WEB" – desktop device• "DOWNLOAD" – downloadable client <p><i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i></p>	Optional
token	<p>Token of the player from Authenticate response.</p> <p><i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i></p>	Optional
win	<p>Win amount in round. Intended to notify Operator about amount won in round.</p> <p>This is notification parameter, and <u>should not be used for transaction within round.</u></p> <p><i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i></p>	Optional
specPrizes[#].specPrizeAmount	<p>number of FRB which was won by player in bingo game (where # - serial number of prize)</p> <p><i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i></p>	Optional (Bingo only)

specPrizes[#].specPrizeCode	unique FRB bonus code within PP system (where # - serial number of prize) <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional (Bingo only)
specPrizes[#].specPrizeType	Type of free special prize (where # - serial number of prize) Possible values: <ul style="list-style-type: none"> “FRB” – free round bonuses <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional (Bingo only)

Response parameters

Name	Description	Status
cash	Real balance of the player.	Required
bonus	Bonus balance of the player.	Required

Example of HTTP request:

```
POST /endRound.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

gameId=vs50hercules&providerId=pragmaticplay&userId=421&roundId=5103579948&platform=DOWNLOAD&hash=0755b1f739655f4d394b20575a7570df
```

Example of JSON response:

```
{
  "cash": 99764.99,
  "bonus": 99.99,
  "error": 0,
  "description": "Success"
}
```


3.12 Refund

Request path: POST /refund.html

Pragmatic Play system may use this method to rollback a bet transaction on the Casino Operator side, in order to reverse the transaction and adjust player's balance. When receive a Refund request Operator have to return money back to player's balance.

Important: The call is idempotent, i.e. sending refund for existing bet again only creates one transaction.

Important: If bet transaction is not found then nothing should happen on the Casino Operator side and success (0) or specific error code for this situation should be returned.

Request parameters

Name	Description	Status
hash	Hash code of the request	Required
userId	Identifier of the user within the Casino Operator's system.	Required
reference	Reference from the original bet transaction.	Required
providerId	Game Provider id.	Required
platform	<p>The platform type (channel) on which the game is played.</p> <p>Possible values:</p> <ul style="list-style-type: none">• "MOBILE" – mobile device• "WEB" – desktop device• "DOWNLOAD" – downloadable client <p><i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i></p>	Optional
amount	<p>Amount to be refunded.</p> <p><i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i></p>	Optional
gameId	<p>Id of the game.</p> <p><i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i></p>	Optional
roundId	<p>Id of the round.</p> <p><i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i></p>	Optional
timestamp	<p>Date and time when the transaction is processed on the Pragmatic Play side</p> <p><i>(Unix epoch time in milliseconds, for example : 1470926696715)</i></p>	Optional

	<i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	
roundDetails	Additional information about the current game round. The field is optional and is not sent to the Operator by default.	Optional
bonusCode	Id of the bonus in Casino Operator system. (*is mandatory in case of FRB API is implemented)	Optional*
	<i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	
token	Token of the player from Authenticate response. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional

Response parameters

Name	Description	Status
transactionId	Id of the refund transaction in Casino Operator system.	Required

Example of HTTP request:

```
POST /refund.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

reference=585c2692f89c56f5ed083692&providerId=pragmaticplay&userId=421&platform=DOWNLOAD&hash=0078aafb64b316a05c91124e4411541a
```

Example of JSON response:

```
{
  "transactionId": "C1482435726839",
  "error": 0,
  "description": "Success"
}
```

3.13 Withdraw

Request path: POST /notify/withdraw.html

Withdraw method is optional and should be implemented only in the case when Operator has requested the downloadable client requiring notifications that player wants to withdraw their money.

When a player finishes playing the games he will click the Withdraw button in the lobby of downloadable client. Pragmatic Play will send a notification to the Operator's system, which will be required to handle the notification, subtract money and return updated player's balance.

Request parameters

Name	Description	Status
hash	Hash code of the request.	Required
userId	Identifier of the user within the Casino Operator's system.	Required
providerId	Game Provider Id.	Required

Response parameters

Name	Description
userId	Identifier of the user within the Casino Operator's system.
currency	Currency of the player.
cash	Real balance of the player.
bonus	Bonus balance of the player.

Example of HTTP request:

```
POST /notify/withdraw.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

userId=421&providerId=pragmaticplay&hash=b4672931ee1d78e4022faadf58e37db
```

Example of JSON response:

```
{
  "userId": "421",
  "currency": "USD",
  "cash": 99999.99,
  "bonus": 99.99,
  "error": 0,
  "description": "Success"
}
```

3.14 GetBalancePerGame

Request path: POST /getBalancePerGame.html

The method allows Pragmatic Play to get player's balance available for certain games. This method is applicable to operators that needs different amount of money to be available in the game client depending on the game type and the policy within Casino Operator system.

Request parameters

Name	Description	Status
hash	Hash code of the request.	Required
userId	Identifier of the user within the Casino Operator's system.	Required
providerId	Game Provider Id.	Required
gameIdList	The list of games for which player's balance should be returned. It is a string contains game id separated by comma.	Required
token	Token of the player from Authenticate response. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional
platform	The platform type (channel) on which the game is played. Possible values: <ul style="list-style-type: none">• "MOBILE" – mobile device• "WEB" – desktop device• "DOWNLOAD" – downloadable client <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional

Response parameters

Name	Description	Status
gamesBalances	The list of player's balances per game. Contains data structures with the following fields: gameId – id of the game cash – Real balance of the player bonus – Bonus balance of the player	Required

Example of HTTP request:

```
POST /getBalancePerGame.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

providerId=...&userId=...&gameIdList=vs20cd,vs20bl,vs7monkeys
```

Example of JSON response:

```
{
  "gamesBalances": [
    { "gameID": "vs20cd", "cash": 25.02, "bonus": 0.00 },
    { "gameID": "vs9c", "cash": 12.02, "bonus": 0.00 }
  ]
}
```

3.15 PromoWin

Request path: POST /promoWin.html

Using this method the Pragmatic Play system will notify Casino Operator about winning that the player is awarded as a result of a campaign that is finished. Notification is asynchronous and may come to the operator with a short delay after the campaign is over. Operator should handle the transaction in their system and send promo win transaction id back to the Pragmatic Play.

Important: The call is idempotent, i.e. sending result again with the same reference number creates only one transaction.

Request parameters

Name	Description	Status
hash	Hash code of the request	Required
providerId	Pragmatic Play provider id in Operator's system.	Required
timestamp	Date and time when the transaction is processed on the Pragmatic Play side (Unix epoch time in milliseconds, for example : 1470926696715)	Required
userId	Id of the player within the operator's system.	Required
campaignId	Id of the campaign.	Required
campaignType	Type of the campaign. Available values are: T – Tournament C – Chat game winning in bingo B – Bonus award	Required
amount	Prize amount that the player is awarded with.	Required
currency	Player's currency.	Required
reference	Unique reference of the transaction within the Pragmatic Play system.	Required

Response parameters

Name	Description	Status
transactionId	Id of the transaction in wallet.	Required
currency	Currency of the player.	Required
cash	Real balance of the player.	Required
bonus	Bonus balance of the player.	Required

Example of HTTP request:

```
POST /promoWin.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

reference=some_reference&campaignId=123456&amount=200.0&providerId=PragmaticPlay&campaignType=T
&userId=some_userId&timestamp=1547473412242&currency=USD&hash=62918d1ba53bee8225c0f2affcf20603
```

Example of JSON response:

```
{
  "transactionId": 1482492905503,
  "currency": "USD",
  "cash": 99815.04,
  "bonus": 99.99,
  "error": 0,
  "description": "Success"
}
```

3.16 SessionExpired

Request path: POST /session/expired

Using this method the Pragmatic Play system will notify Casino Operator that player's session has expired in Pragmatic Play system due to long inactivity or game closing.

The method is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this method to be sent, they should ask the Pragmatic Play's Technical Support for additional configuration

Request parameters

Name	Description	Status
hash	Hash code of the request	Required
providerId	Pragmatic Play provider id in Operator's system.	Required
sessionId	Player's game session id on Pragmatic Play side	Required
playerId	Id of the player within the operator's system.	Required
token	Token of the player from Authenticate response. <i>The field is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	Optional

Response parameters

Name	Description	Status
error	Response status.	Required
description	Response status short description.	Required

Example of HTTP request:

```
POST /cgAPItest/v3/httpService/session/expired HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded

providerId=PragmaticPlay&sessionId=6fd2d6f3bb8f4c5a9fadf15d81206af2&playerId=123456&hash=c46d5b113e81d306831a06765e12067f
```

Example of JSON response:

```
{
  "error": 0,
  "description": "Success"
}
```

3.17 Error codes

Below table contains the error codes that should be returned by Casino Operator in the response to Seamless Wallet API calls.

Code	Description	Send Reconciliation call	
		Bet (send Refund)	Result/Refund etc. (send retry)
0	Success.	No	No
1	Insufficient balance. The error should be returned in the response on the Bet request.	No	Yes
2	Player not found or is logged out. Should be returned in the response on any request sent by Pragmatic Play if the player can't be found or is logged out at Casino Operator's side.	Yes	Yes
3	Bet is not allowed. Should be returned in any case when the player is not allowed to play a specific game. For example, because of special bonus.	No	Yes
4	Player authentication failed due to invalid, not found or expired token.	Yes	Yes
5	Invalid hash code. Should be returned in the response on any request sent by Pragmatic Play if the hash code validation is failed.	Yes	Yes
6	Player is frozen. Casino Operator will return this error in the response of any request if player account is banned or frozen.	Yes	Yes
7	Bad parameters in the request, please check post parameters.	Yes	Yes
8	Game is not found or disabled. This error should be returned on Bet request if the game cannot be played by some reason. Bet result request with winning amount should be processed as intended, even if the game is disabled.	Yes	Yes
50	Bet limit has been reached. The code is relevant for regulated markets.	No	Yes
100	Internal server error. Casino Operator will return this error code if their system has internal problem and cannot process the request at the moment and Operator logic requires a retry of the request. <u>Request will follow Reconciliation process</u>	Yes	Yes
120	Internal server error. Casino Operator will return this error code if their system has internal problem and cannot process the request and Operator logic does not require a retry of the request. <u>Request will NOT follow Reconciliation process</u>	No	No
130	Internal server error on EndRound processing. Casino Operator will return this error code if their system has internal problem and cannot process the EndRound request, and Operator logic <u>requires</u> a retry of the request. <u>This error code should be used for Endround method only and not for other methods</u>	-	-
210	Reality check warning	Yes	Yes
310	Player's bet is out of his bet limits. Should be returned if player's limits have been changed, and the bet is out of new limit levels. Game client will show a proper error message, and ask player to reopen the game. After game reopening new bet limits will be applied. The error is relevant for operators that send player's bet limits in response on Authenticate request.	No	No

IV. Balance Transfer API

This is a simple API for game Operator to connect to the Pragmatic Play game system. API is an HTTP listener, which listens POST requests coming to an URL with the request mappings below.

All responses should be in JSON format and should contain error code and error description, which is empty in the case if no error occurs.

URL of the HTTP service will be provided by Pragmatic Play for the production and test environments and looks like:

`https://{API service domain}/IntegrationService/v3/http/CasinoGameAPI`

Balance Transfer HTTP service is securely protected, hence please be sure:

- player's browser (end) must NOT be used as initiator of requests to the service;
- proper IPs are supplied to Pragmatic Play for whitelisting

4.1 Game opening

After Operator calls StartGame method at Pragmatic Play side he gets an URL for game opening.

The link will look like:

`https://{game server domain}/gs2c/openGame.do?tc={ticket}&stylename={secureLogin}`

where:

- game server domain – a domain name for game opening and playing;
- ticket – secure one-time token generated by Pragmatic Play.

4.2 Hash calculation

Hash code is calculated with following formula:

From request post parameters, all parameters are taken (except hash) and appending to string:

1. Sort all parameter by keys in alphabetical order.
2. Append them (if the value is not null or empty) in key1=value1&key2=value2.
3. Append secret key, e.g.: key1=value1&key2=value2SECRET.
4. Calculate the hash by using MD5.
5. Compare with hash parameter. In the case of failure Casino Operator should send the error code 5.

4.3 Data types

Name	Type	Description
externalPlayerId	String(100)	Unique identifier of the player within the Casino Operator system. Parameter value is case-sensitive. <i>Example: joe1001, 2644987, playerABC*, playerAbc*</i> * playerABC and playerAbc – are two different player accounts within Pragmatic Play system
playerId	String(100)	Unique identifier of the player within the Pragmatic Play system. May be used for troubleshooting. <i>Example: 1000, 2644987</i>
currency	String(3)	Currency of the player, 3-character ISO 4217 code. <i>Example: USD, EUR</i>
language	String(2)	Language code in ISO 639-1 standard (e.g. en, fr, it).
externalTransactionId	String(100)	Unique identifier of the transaction within Casino Operator system. <i>Example: 5623489, CHR789452, c0304c72-514f-47a4-8ae1-eb755224391b</i>
transactionId	String(100)	Unique identifier of the transaction within Pragmatic Play system. <i>Example: 5623489, CHR789452, c0304c72-514f-47a4-8ae1-eb755224391b</i>
amount	Decimal (10, 2)	Amount that is transferred in or out of the player's balance. Can have positive (deposit) or negative (withdrawal) values. <i>Example: 20.35, -40.00</i>
balance	Decimal (10, 2)	Player's current balance. Example: '100.00'.
gameId	String(20)	Symbolic unique identifier of the game within the Pragmatic Play system. <i>Example: vs25queenofgold, vs50aladdin, cs5triple8gold, bjma, rla, vpa</i> The list of available games may be obtained by Casino Operator by calling GetCasinoGames method of CasinoGameAPI of <i>Integration simple HTTP Service</i> .
platform	String(20)	Platform for which the game should be opened. Available values: <ul style="list-style-type: none">• MOBILE – if game should be opened on mobile devices• WEB – if game will be opened on desktop devices If this parameter is not defined for game opening request, desktop version of the game will be opened by default. The list of platforms, supported by specific game, can be found in Game Library.
URL	String(250)	URL of a certain resource on the Casino Operator or Pragmatic Play website. LobbyUrl, CashierURL – are the links to Casino Operator's resources that will be opened when player clicks "Fund" or "Home" button in the game. GameURL – is the link for opening a game on Pragmatic Play's game server.
error	Integer	0 - if the request was processed successfully or error code in other case.
description	String (250)	Description of the error for troubleshooting purpose.

4.4 CreatePlayer

Request path: POST /player/account/create/

Using this method Casino Operator will create player's account within the Pragmatic Play system. This method should be called before player is sent to the Pragmatic Play's games.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
externalPlayerId	Id of the player within the Operator system.	Required
currency	Currency of the player.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
playerId	Id of the player within the Pragmatic Play system.
Error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/player/account/create/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&externalPlayerId=test_player&currency=USD&hash=620a4016ae32ea0efbebfef1b3544571c
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "playerId": 6749178
}
```

4.5 Transfer

Request path: POST /balance/transfer/

This method transfers funds in to player's balance (i.e. deposit) and out of the player's balance (i.e. withdrawal) within Pragmatic Play system. Positive amount of money indicates fund in and negative amount means fund out transaction.

Important: In cases where transfer call fails due to:

- **network error (HTTP status \neq 200)**
- **error: 1 (description: "internal error")** in response

Operator should send idempotent retry calls (with the same externalTransactionId).
Recommended actions are specified in 4.10 Error codes

Important: The call is idempotent, i.e. sending it again only creates one transaction.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
externalPlayerId	Id of the player within the Casino Operator system.	Required
externalTransactionId	Id of the transaction within Casino Operator system.	Required
amount	Amount to be added to player's balance (positive value) or to be subtracted from player's balance (negative value), in player's currency.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
transactionId	Id of the transfer transaction within the Pragmatic Play system.
balance	Player's balance after successful transaction.
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/balance/transfer/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&externalPlayerId=test_player&externalTransactionId=EXT123456789&hash=380720e974ae2b775ce933fb73eb8fcf&amount=999.99
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "transactionId": 1908759,
  "balance": 999.99
}
```

}

4.6 GetTransferStatus

Request path: POST /balance/transfer/status/

This method returns the status of a particular transaction that transferred the money in or out the player's balance on the Pragmatic Play side.

Important: This method will return player's balance after successful transaction for all transactions that was created not earlier than August 3, 2017. For older transactions player's balance in the response will be null. Description field of the response will contain the following text: "Transaction is older than 3.08.2017".

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
externalTransactionId	Id of the transaction within Casino Operator system.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
transactionId	Id of the transfer transaction within the Pragmatic Play system.
status	Status of the transaction: <ul style="list-style-type: none">○ 'Success' – The transaction was successfully processed○ 'Not found' – The transaction was not found (was not processed)
amount	Amount added to player's balance (positive value) or subtracted from player's balance (negative value), in player's currency.
balance	Player's balance after successful transaction. <i>Note: The field will contain current player's balance in case the transaction was not processed.</i>
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/balance/transfer/status/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&externalTransactionId=EXT123456789&hash=380720e974ae2b775ce933fb73eb8fcf
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "transactionId": 1908759,
  "status": "Success",
  "amount": "999.99",
  "balance": 999.99
}
```

4.7 GetBalance

Request path: POST /balance/current/

Using this method Operator can get the current balance of the player in the Pragmatic Play system.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
externalPlayerId	Id of the player within the Casino Operator system.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
balance	Current player's balance within the Pragmatic Play system.
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/balance/current/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

externalPlayerId=test_player&secureLogin=username&hash=405281156fc4cace4d2385a8ead83290
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "balance": 999.99
}
```

4.8 StartGame

Request path: POST /game/start/

Using this method Operator can receive a valid URL to requested game. A specific ticket will be generated for player defined in the request for a particular game. This ticket will be verified on the Pragmatic Play side when player opens a game.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
externalPlayerId	Id of the player within the Operator system.	Required
gameId	Id of the game within the Pragmatic Play system.	Required
language	Language on which the game should be opened.	Required
platform	Platform for which the game should be opened	Optional
cashierURL	Link for opening the cashier on Casino Operator site when a player has no funds	Optional
lobbyURL	Link for returning to Lobby page on Casino Operator site. This link is used for <i>Back to Lobby (Home)</i> button in mobile version of games	Optional
hash	Hash code of the request.	Required

Response parameters

Name	Description
gameURL	Link to the game to be open on the Pragmatic Play side. This link will contain special parameters which should remain as is.

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/game/start/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&externalPlayerId=test_player&gameId=vs50aladdin&language=en&platform=WEB&cashierURL=http%3A%2F%2Fdomain.com%2F&lobbyURL=http%3A%2F%2Fdomain.com%2F&hash=438fea0a68a0e9ab49df756cc20eda8f
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "gameURL":
  "https://{game_server_domain}/gs2c/openGame.do?tc=SHq6rK8JEUJRKmeAqD3Ceim1Wojtma1MA7xsu4ciynHFWXosjDjgAUPH1A3FSoAM&stylename=username&lang=en"
}
```


4.9 TerminateSession

Request path: POST /game/session/terminate/

Using this method Operator can terminate the current player session and kick him out all games.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
externalPlayerId	Id of the player within the Casino Operator system.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/CasinoGameAPI/game/session/terminate/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&externalPlayerId=test_player&hash=405281156fc4cace4d2385a8ead83290
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK"
}
```

4.10 Error codes

Below table contains the error codes that the Pragmatic Play system will return in the response to Balance Transfer API calls.

Only responses with **HTTP Status: 200** should be accepted by operator as valid response

Responses with HTTP Status other than 200 **should not be accepted** as valid (recommended actions are the same as for **error: 1, description "Internal error. Try later please."**)

For each API account, retry frequency for failed "Transfer" or "GetTransferStatus" method requests, should be not more than **one call per minute** and **not longer than 1 hour**. (after 1 hour Error reason should be checked in relevant support channel)

Code	Description	Further action recommendations	
		Transfer method	Other methods
1	Internal error. Try later please.	Options: 1. GetTransferStatus method retries After getting valid response: - Transfer success – no action required - Transfer failed – Transfer transaction failed, amount can be returned to player's wallet 2. Transfer call retry After getting valid response: - Transfer success (error code 0 or 8) – no action required - Transfer failed – Transfer transaction failed, amount can be returned to player's wallet	Error reason should be checked in relevant support channel
	Insufficient funds available to complete the transaction.	GetBalance request, to get actual balance	-
2	Incorrect secure LOGIN and secure password combination (will be returned also if the hash code validation is failed)	Verify request values	Verify request values
6	Game is not found or is not allowed for your system	-	Error cause should be checked in relevant support channel
7	One or several input parameters is not set or set incorrectly.	Verify request values	Verify request values
8	Transaction already exists	No action required	-
17	Player not found	Verify request values	Verify request values
21	Currency code is incorrect or unsupported	-	Verify request values

V. Simple Free Rounds API

Free Rounds Bonus (FRB) allows the player to play a specified number of free rounds in a game. Free rounds are always played with a minimal bet specified for each game and currency. During FRB game play, bets are not deducted from the player's balance, but all wins are collected for the FRB in the Pragmatic Play's system.

After all free rounds are played the Pragmatic Play system sends a bonusWin API call to Operator server to add money to the player's balance. Please note that not all games support FRB.

Free Rounds Bonus (FRB) can be awarded to players registered within the Pragmatic Play system using API. If a player is not registered in the Pragmatic Play system yet a new player account will be created automatically.

Free Rounds Bonus HTTP service is a simple API for game Operator to manage Free Rounds Bonuses. API is an HTTP listener, which listens POST requests coming to an URL with the request mappings below.

All responses should be in JSON format.

URL of the FRB HTTP service will be provided by Pragmatic Play for the production and test environments and looks like:

`https://{API service domain}/IntegrationService/v3/http/FreeRoundsBonusAPI`

Free Rounds Bonus HTTP service is securely protected, hence please be sure:

- player's browser (end) must NOT be used as initiator of requests to the service;
- proper IPs are supplied to Pragmatic Play for whitelisting

5.1 Hash calculation

Hash code is calculated with following formula:

From request post parameters, all parameters are taken (except hash) and appending to string:

1. Sort all parameter by keys in alphabetical order.
2. Append them (if the value is not null or empty) in key1=value1&key2=value2.
3. Append secret key, e.g.: key1=value1&key2=value2SECRET.
4. Calculate the hash by using MD5.
5. Compare with hash parameter. In the case of failure Casino Operator should send the error code 5.

5.2 Data types

Name	Type	Description
playerId	String(100)	Id of the player within the Operator system. Examples: 1000, 2644987.
currency	String	Player's ISO 4217 currency code. Examples: 'EUR' or 'USD'.
gameId	String	Unique identifier of the game.
rounds	Integer	Number of free rounds
bonusCode	String	Bonus unique identifier within the Operator system.
expirationDate, validityDate	Long	Date and time when the bonus will expire - timestamp in seconds. (Unix epoch time in seconds, for example : 1470926605)
error	Integer	Error code.

5.3 CreateFRB

Request path: POST /FreeRoundsBonusAPI/createFRB/

Using this method Casino Operator system can create a Free Round Bonus in the Pragmatic Play system.

In case of a network error Operator can send a repeated request. The Pragmatic Play system will not create new RF bonus if there is an active FR bonus with the same bonus code.

If specified player account is not found in the Pragmatic Play system, it will be registered automatically.

Note: Expiration date is mandatory. It should be sent as timestamp in seconds, and the bonus validity period must not be longer than 30 days.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
playerId	Id of the player within the Casino Operator system.	Required
currency	Currency of the player.	Required
gameIDList	List of symbolic unique identifiers of the game that the FR is awarded for, comma separated. Example: vs25a, vs9c, vs20s.	Required
rounds	Number of free game rounds awarded.	Required
bonusCode	Bonus id within the Casino Operator system. Should be unique within the brand.	Required
expirationDate	Date and time when the free rounds will expire.	Required
validityDate	Date and time when the bonus gets invalid and is unavailable for the player.	Optional
hash	Hash code of the request.	Required

Response parameters

Name	Description
providerPlayerId	Id of the player within the Pragmatic Play system.
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/FreeRoundsBonusAPI/createFRB HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&playerId=421&currency=USD&gameIDList=vs20b1&rounds=10&bonusCode=421_frb&expirationDate=1482588510&hash=1d319398565e48dc635c3c621475d0ac
```

Example of JSON response:

```
{
  "error": "0",
  "description": "",
}
```

```
"providerPlayerID": 6638030
}
```

5.4 CancelFRB

Request path: POST /FreeRoundsBonusAPI/cancelFRB/

Using this method Casino Operator system can cancel a Free Round Bonus in the Pragmatic Play system. Free Rounds will be canceled in the Pragmatic Play system regardless of whether the player started to play them or not.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
bonusCode	Bonus id within the Operator system. Should be unique within the brand.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/FreeRoundsBonusAPI/cancelFRB HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&bonusCode=421_frb&hash=39554fed4f41132eb8fe75e9a7ba3df6
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK"
}
```

5.5 GetPlayersFRB

Request path: POST /FreeRoundsBonusAPI/getPlayersFRB/

Using this method Casino Operator system can get from the Pragmatic Play system all Free Round Bonuses that are awarded to the particular player.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
playerId	Id of the player within the Operator system.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
bonuses	List of information about all active FRB that were awarded to the player. See below FRB type description of the objects in the list.
error	Error code.
description	Description of the error for troubleshooting.

Type: FRB

Name	Description
currency	Currency of the player.
gameIDList	List of symbolic unique identifiers of the game that the FR is awarded for, comma separated. Example: vs25a, vs9c, vs20s.
rounds	Number of free game rounds awarded.
roundsPlayed	Number of rounds already played.
bonusCode	Bonus id within the Operator system. Should be unique within the brand.
expirationDate	Date and time when the free rounds will expire.

Example of HTTP request:

```
POST /IntegrationService/v3/http/FreeRoundsBonusAPI/getPlayersFRB HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&playerId=421&hash=b623382e6dd3219d63b7ed1979408856
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "bonuses": [
    {
      "currency": "USD",
      "gameIDList": "vs20b1",
      "rounds": 10,
      "roundsPlayed": 0,
      "bonusCode": "421_frb1",
      "expirationDate": "2016-12-24 00:00",
    },
    {
      "currency": "USD",
      "gameIDList": "vs20b1",
      "rounds": 10,
      "roundsPlayed": 0,
      "bonusCode": "421_frb2",
      "expirationDate": "2016-12-24 00:00",
    }
  ]
}
```

5.6 Error codes

The table below contains error codes that the Pragmatic Play system will return in the response to Simple Free Rounds API calls.

Code	Description
0	Request was successfully processed.
1	Authentication failed. Incorrect secure login and secure password combination.
2	Validation failed. Empty mandatory field '{field name}'.
3	Game(s) are not supported: {list of the game id}.
4	Game(s) do not support Free round bonus: {list of the game id}
5	Bonus code already exists.
6	The requested Free Round bonus is not found in the system.
7	Free round bonus is canceled.
8	Free round bonus is closed or started to play.
9	Currency code '{ISO code}' is incorrect or unsupported.
10	Player does not have active Free round bonuses.
11	Free round bonus cannot be created. Expiration date limit (30 days) has been exceeded.
12	Free round bonus cannot be created. Expiration date is in the past.
14	Bonus code already exists with another parameters.
1000	Internal server error. Pragmatic Play will return this error code if their system has internal problem and cannot process the request.

VI. Variable Free Rounds API

Free Rounds Bonus (FRB) allows the player to play a specified number of free rounds in a game. Free rounds created via Variable Free Rounds API may be configured to be played with a certain bet value specified for each game and currency. During Free rounds game play, bets are not deducted from the player's balance, but all wins are collected for the Free Round Bonus in the Pragmatic Play's system.

After all free rounds are played the Pragmatic Play system sends a bonusWin API call to Operator server to add money to the player's balance. Please note that not all games support FRB.

Free rounds can be awarded to players registered within the Pragmatic Play system using API. If a player is not registered in the Pragmatic Play system yet a new player account will be created automatically.

Free Rounds Bonus HTTP service is a simple API for game Operator to manage Free Rounds. API is an HTTP listener, which listens POST requests coming to an URL with the request mappings below.

All responses should be in JSON format.

URL of the FRB HTTP service will be provided by Pragmatic Play for the production and test environments and looks like: *https://{API service domain}/IntegrationService/v3/http/FreeRoundsBonusAPI/v2/*

Free Rounds HTTP service is securely protected, hence please be sure:

- player's browser (end) must NOT be used as initiator of requests to the service;
- proper IPs are supplied to Pragmatic Play for whitelisting

6.1 Hash calculation

Hash code is calculated with following formula:

From request post parameters, all parameters are taken (except hash) and appending to string:

6. Sort all parameter by keys in alphabetical order.
7. Append them (if the value is not null or empty) in key1=value1&key2=value2.
8. Append secret key, e.g.: key1=value1&key2=value2SECRET.
9. Calculate the hash by using MD5.
10. Compare with hash parameter. In the case of failure Casino Operator should send the error code 5.

6.2 Data types

Name	Type	Description
playerId	String(100)	Id of the player within the Operator system. Examples: 1000, 2644987.
currency	String	Player's ISO 4217 currency code. Examples: 'EUR' or 'USD'.
gameId	String	Unique identifier of the game.
rounds	Integer	Number of free rounds
bonusCode	String	Bonus unique identifier within the Operator system.
expirationDate, startDate, validityDate	Long	Date and time when the bonus will start, expire or until it is valid - timestamp in seconds. (Unix epoch time in seconds, for example : 1470926605)
error	Integer	Error code.

6.3 Create Free Rounds

Request path: POST /FreeRoundsBonusAPI/v2/bonus/create/

Using this method Operator can create a new Free Rounds Bonus with the Pragmatic Play's games.

In case of a network error Operator can send a repeated Create Free Rounds requests. The Pragmatic Play system will not create new RF bonus if there is an active FR bonus with the same bonus code.

Note:

1. Expiration date is mandatory. It should be sent as timestamp in seconds, and the bonus validity period must not be longer than 30 days.
2. Total bet for each round is calculated as bet-per-line * number-of-lines.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
bonusCode	Bonus id within the Casino Operator's system. Should be unique within the brand.	Required
startDate	Date and time when free rounds bonus will start.	Required
expirationDate	Date and time when the free rounds will expire.	Required
validityDate	Date and time when the bonus gets invalid and is unavailable for the player.	Optional
type	Type of the Free Rounds Bonus that the player will be awarded. Available values are: S – Free rounds with the predefined number of spins	Optional
rounds	Number of free game rounds awarded.	Optional
hash	Hash code of the request.	Required
gameList	List of the games associated with the bonus, with bet per line values for each currency. See below VariableFRGame type description of the objects in the list. <i>This data must be sent as a JSON in the body of the request.</i>	Required

Type: VariableFRGame

Name	Description
gameId	Id of the game associated with the free rounds bonus.
betValues	List of bets per line with the currency.

Type: VariableFRBetValue

Name	Description
currency	Currency code.
betPerLine	Value of bet per line in the specified currency, which will be applied during free rounds

Response parameters

Name	Description
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/FreeRoundsBonusAPI/v2/bonus/create?bonusCode=Bonus_Code&expirationDate=1540563996&rounds=27&secureLogin=Secure_Login&startDate=1537971996&validityDate=1543242396&hash=464e9816b531cc78bbdcffc81dd44e18 HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/json
```

```
{
  "gameList": [
    {
      "gameId": "vs25a",
      "betValues": [
        {"betPerLine": 0.05, "currency": "USD"},
        {"betPerLine": 0.10, "currency": "EUR"},
        {"betPerLine": 0.50, "currency": "CNY"}
      ]
    },
    {
      "gameId": "vs9c",
      "betValues": [
        {"betPerLine": 0.15, "currency": "USD"},
        {"betPerLine": 0.30, "currency": "EUR"},
        {"betPerLine": 1.50, "currency": "CNY"}
      ]
    }
  ]
}
```

Example of JSON response:

```
{
  "error": "0",
  "description": ""
}
```

6.4 Cancel Free Rounds

Request path: POST /FreeRoundsBonusAPI/v2/bonus/cancel/

Using this method Operator can cancel an existing Free Rounds Bonus in the Pragmatic Play system. Free Rounds will be canceled in the Pragmatic Play system regardless of whether the player started to play them or not.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
bonusCode	Bonus id within the Operator system. Should be unique within the brand.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/FreeRoundsBonusAPI/v2/bonus/cancel HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&bonusCode=421&hash=39554fed4f41132eb8fe75e9a7ba3df6
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK"
}
```

6.5 Add players

Request path: POST /FreeRoundsBonusAPI/v2/players/add/

Using this method Operator can add players to the existing Free Round Bonus. If player's account does not exist in the Pragmatic Play system yet, it will be registered automatically.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
bonusCode	Bonus id within the Operator system. Should be unique within the brand.	Required
playerList	List of player id to add to the existing Free Round Bonus, comma separated. This data must be sent as a JSON in the body of the request.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST
/IntegrationService/v3/http/FreeRoundsBonusAPI/v2/players/add/?secureLogin=ext_test1&bonusCode=421&
hash=908ea6c3335602d153e490871b376c77 HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/json

{
  "playerList": ["449986","450013","450509","437070"]
}
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK"
}
```

6.6 Remove players

Request path: POST /FreeRoundsBonusAPI/v2/players/remove/

Using this method Operator can remove players from the existing Free Round Bonus and cancel free rounds awarded to them. The bonus will be canceled for a particular player even in case the player had started playing free rounds.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
bonusCode	Bonus id within the Operator system. Should be unique within the brand.	Required
playerList	List of player id to remove from the existing Free Round Bonus, comma separated.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/FreeRoundsBonusAPI/v2/players/remove HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&bonusCode=421&hash=39554fed4f41132eb8fe75e9a7ba3df6

{
  "playerList": ["449986","450013","450509","437070"]
}
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK"
}
```

6.7 Create Player Free Rounds

Request path: POST /FreeRoundsBonusAPI/v2/bonus/player/create/

Using this method Operator can create a new Free Rounds Bonus with the Pragmatic Play's games for player

In case of a network error Operator can send a repeated Create Free Rounds requests. The Pragmatic Play system will not create new RF bonus if there is an active FR bonus with the same bonus code.

Note:

- Expiration date is mandatory. It should be sent as timestamp in seconds, and the bonus validity period must not be longer than 30 days.
- Total bet for each round is calculated as bet-per-line * number-of-lines.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
bonusCode	Bonus id within the Casino Operator's system. Should be unique within the brand.	Required
startDate	Date and time when free rounds bonus will start.	Required
expirationDate	Date and time when the free rounds will expire.	Required
validityDate	Date and time when the bonus gets invalid and is unavailable for the player.	Optional
rounds	Number of free game rounds awarded.	Optional
playerId	Player ID in Casino Operator's system.	Required
currency	Player currency.	Required
hash	Hash code of the request.	Required
gameList	List of the games associated with the bonus, with bet per line values for each currency. See below VariableFRGame type description of the objects in the list. <i>This data must be sent as a JSON in the body of the request.</i>	Required

Type: VariableFRGame

Name	Description
gameId	Id of the game associated with the free rounds bonus.
betValues	List of bets per line with the currency.

Type: VariableFRBetValue

Name	Description
currency	Currency code.
betPerLine	Value of bet per line in the specified currency, which will be applied during free rounds

Response parameters

Name	Description
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/FreeRoundsBonusAPI/v2/bonus/player/create/?bonusCode=some_bomnus_c
ode123&expirationDate=1597921169&rounds=27&secureLogin=some_secureLogin&startDate=1597835415&curren
cy=USD&playerId=392&hash=2b7e7dc8ef0ecd2dcd9c89b9f79fbae1 HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/json
```

```
{
  "gameList": [
    {
      "gameId": "vs25a",
      "betValues": [
        {"betPerLine": 0.05, "currency": "USD"},
        {"betPerLine": 0.10, "currency": "EUR"},
        {"betPerLine": 0.50, "currency": "CNY"}
      ]
    },
    {
      "gameId": "vs9c",
      "betValues": [
        {"betPerLine": 0.15, "currency": "USD"},
        {"betPerLine": 0.30, "currency": "EUR"},
        {"betPerLine": 1.50, "currency": "CNY"}
      ]
    }
  ]
}
```

Example of JSON response:

```
{
  "error": "0",
  "description": ""
}
```


6.8 Error codes

The table below contains the error codes that the Pragmatic Play system will return in the response to Variable Free Rounds API calls.

Code	Description
0	Request was successfully processed.
1	Authentication failed. Incorrect secure login and secure password combination.
2	Validation failed. Empty mandatory field '{field name}'.
3	Game(s) are not supported: {list of the game id}.
4	Game(s) do not support Free round bonus: {list of the game id}
5	Bonus code already exists.
6	The requested Free Round bonus is not found in the system.
7	Free round bonus is canceled.
8	Free round bonus is closed or started to play.
9	Currency code '{ISO code}' is incorrect or unsupported.
10	Player does not have active Free round bonuses.
11	Free round bonus cannot be created. Expiration date limit (30 days) has been exceeded.
12	Free round bonus cannot be created. Expiration date is in the past.
14	Bonus code already exists with another parameters.
1000	Internal server error. Pragmatic Play will return this error code if their system has internal problem and cannot process the request.

VII. Game History API

History API provides game rounds played by the player, with their details. Operator can get a list of games played, game rounds that the player has played during a particular day and hour, and the screen of the game at the end of the game round.

History HTTP service is a simple API for game Operator to get player's game history. API is an HTTP listener, which listens POST requests coming to an URL with the request mappings below.

All responses should be in JSON format.

URL of the history HTTP service will be provided by Pragmatic Play for the production and test environments and looks like:

`https://{API service domain}/IntegrationService/v3/http/HistoryAPI/`

History HTTP service is securely protected, hence please be sure:

- player's browser (end) must NOT be used as initiator of requests to the service;
- proper IPs are supplied to Pragmatic Play for whitelisting

7.1 Hash calculation

Hash code is calculated with following formula:

From request post parameters, all parameters are taken (except hash) and appending to string:

11. Sort all parameter by keys in alphabetical order.
12. Append them (if the value is not null or empty) in key1=value1&key2=value2.
13. Append secret key, e.g.: key1=value1&key2=value2SECRET.
14. Calculate the hash by using MD5.
15. Compare with hash parameter. In the case of failure Casino Operator should send the error code 5.

7.2 Data types

Name	Type	Description
secureLogin	String (32)	User name for authentication in the Casino Game API service. Should be provided by the Pragmatic Play.
playerId	String(100)	Id of the player within the Operator system. Examples: 1000, 2644987.
gameId	String (32)	Symbolic unique identifier of the game within the Pragmatic Play system. <i>Example: vs25queenofgold, vs50aladdin, cs5triple8gold, bjma, rla, vpa</i> Casino Operator can get the list of all available games by a call of GetCasinoGames method of <i>Integration API</i> .
language	String(2)	Language code in ISO 639-1 standard (e.g. en, fr, it).
roundId	Long	Identification for the player's round.
currency	String(3)	Player's ISO 4217 currency code. Examples: 'EUR' or 'USD'.
cash	Decimal (10, 2)	Player's real balance. Example: '100.00'.
bonus	Decimal (10, 2)	Player's promo balance. Example: '100.00'.
betAmount	Decimal (10, 2)	Amount of the bet. Example: '1.00'.
winAmount	Decimal (10, 2)	Amount of the win. Example: '5.25'.
roundDetails	String(50)	Additional information about the current game round, such as "Free spin", "Bonus", etc.
datePlayed	DateTime	Date and time when the round was played
error	Integer	Error code.

7.3 GetPlayedGames

Request path: POST /GetPlayedGames/

Using this method Casino Operator can get a list of the games played by the player during the day.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service.	Required
playerId	Id of the player within the Casino Operator system.	Required
datePlayed	Date, based on the time zone of the user.	Required
timeZone	Time zone of the user. <i>Example: GMT, GMT+8, GMT+04:00</i>	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
games	List of games played. See below GamePlayed type description of the objects in the list.

Type: GamePlayed

Name	Description
gameId	Symbolic unique identifier of the game.
gameName	Name of the game.
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/HistoryAPI/GetPlayedGames/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&playerId=421&datePlayed=2016-12-23&timeZone=GMT+00:00&hash=980d06d3361f1e21a2f1550c6806ef52
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "games": [
    {
      "gameId": "vs30catz_jp",
      "gameName": "The Catfather Part II"
    },
    {
      "gameId": "vs50aladdin",
      "gameName": "3 Genie Wishes"
    }
  ]
}
```

7.4 GetGameRounds

Request path: POST /GetGameRounds/

Using this method Casino Operator can get a list of the game rounds played by the player during the certain day and (optionally) the specific hour.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service.	Required
playerId	Id of the player within the Casino Operator system.	Required
datePlayed	Date when the game rounds were played, based on time zone of the user.	Required
timeZone	Time zone of the user. <i>Example: GMT, GMT+8, GMT+04:00</i>	Required
gameId	Symbolic unique identifier of the game.	Required
hour	A number of hour then rounds were played (optional).	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
rounds	List of game rounds played. See below GameRound type description of the objects in the list.
error	Error code.
description	Description of the error for troubleshooting.

Type: GameRound

Name	Description
dateTime	Date and time when the round was played, based on time zone of the user. The value is returned in Pragmatic Play's server time zone (UTC/GMT+0).
gameId	Symbolic unique identifier of the game.
gameName	Name of the game.
roundId	Unique identifier of the game round.
currency	Player's currency.
betAmount	Bet amount.
winAmount	Win amount.
balance	Player's balance at the end of the round.
roundDetails	Additional information about the game round, such as "Free spin", "Bonus", etc.
memberId	Unique id for player on the Pragmatic Play side.
casinoID	Unique id for brand on the Pragmatic Play side.
detailsUrl	This parameter is not currently used and should be ignored by operators

roundStatus	<p>Status of the game round. May have the following values:</p> <ul style="list-style-type: none"> ○ In progress – game round was started but not finished yet by the player ○ Completed – game round has been completed by the player ○ Cancelled – game round has been closed automatically by the game round finalization process ○ CompleteInProcess – game round is marked as Completed in the db; BetResult or EndRound requests is in asynchronous transaction queue and the system tries to send it to Operator <p>CancelInProcess – game round is marked as Cancelled in the db; Refund is in asynchronous queue and being sent to Operator.</p>
-------------	---

Example of HTTP request:

```
POST /IntegrationService/v3/http/HistoryAPI/GetGameRounds/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&playerId=421&datePlayed=2016-12-23&timeZone=GMT+00:00&gameId=vs50aladdin&hour=10&hash=0179f75a37274337f9aefe2c3108db0f
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "rounds": [
    {
      "dateTime": "2016-12-23 05:50:35.0",
      "gameId": "vs50aladdin",
      "gameName": "3 Genie Wishes",
      "roundId": "5108924498",
      "currency": "USD",
      "betAmount": "0.0",
      "winAmount": "0.05",
      "balance": "99711.59",
      "roundDetails": "Free spin",
      "memberId": 6638030,
      "detailsUrl": "pragmaticplay"
    }, ... ,
    {
      "dateTime": "2016-12-23 05:54:28.0",
      "gameId": "vs50aladdin",
      "gameName": "3 Genie Wishes",
      "roundId": "5108946371",
      "currency": "USD",
      "betAmount": "2.5",
      "winAmount": "0.0",
      "balance": "99710.04",
      "roundDetails": null,
      "memberId": 6638030,
      "detailsUrl": "pragmaticplay"
    }
  ]
}
```

7.5 OpenHistory

Request path: POST /OpenHistory/

Using this method Casino Operator can get a link to the page with the game details at the end of the round. The method returns URL that contains round id and a one-time password that the game server will use for validation of the request, for the security reason.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service.	Required
playerId	Identifier of the player within the Casino Operator system.	Required
roundId	Unique identifier of the game round.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
historyUrl	URL for opening the game screen page.
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/HistoryAPI/OpenHistory/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&playerId=421&roundId=5108924498&hash=361527d01b040fd0c5fb68dcc204b707
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "url":
  "https://{game_server_domain}/gs2c/gameHistoryDetailForApi.do?otp=9KngwmdYVw6YLViPVIjU9eEMFs1z1ALnw
toZygNWTMB7SqAApl8ERLysn7Xp26p0&playSessionId=5108924498"
}
```

7.6 GetRoundStatus

Request path: POST /GetRoundStatus/

Method returns the current status of a particular game round. Using this method Casino Operator can check game rounds recorded on their side and cancel open bets returning money to the player's balance.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service.	Required
roundId	Id of the game round.	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
roundId	Id of the game round.
betAmount	Amount of the bet.
winAmount	Amount of the winnings.
roundStatus	Status of the game round. May have the following values: <ul style="list-style-type: none">○ In progress – game round was started but not finished yet by the player○ Completed – game round has been completed by the player○ Cancelled – game round has been closed automatically by the game round finalization process○ CompleteInProcess – game round is marked as Completed in the db; BetResult or EndRound requests is in asynchronous transaction queue and the system tries to send it to Operator○ CancelInProcess – game round is marked as Cancelled in the db; Refund is in asynchronous queue and being sent to Operator.
error	Error code.
description	Description of the error for troubleshooting.

7.7 OpenHistoryExtended

Request path: POST /OpenHistoryExtended/

Using this method Casino Operator can obtain detailed information about played game round, including free spin sessions.

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service.	Required
playerId	Id of the player within the Operator system.	Required
gameId	Id of the game.	Required
roundId	Unique identifier of the game round.	Required
language	Language of report page	Optional
hash	Hash code of the request.	Required

Response parameters

Name	Description
url	Report page URL.
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /IntegrationService/v3/http/HistoryAPI/OpenHistoryExtended/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&playerId=123123&gameId=vs7monkeys&roundId=111222&hash=5fd31717ba4da9e78ac19730212925c2
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK"
  "url": "https://{game server domain}/gs2c/parentRoundHistoryDetails.do?playSessionId=111222&style
name=username&hash=9cf3e7442c8ea5e95c22d123500ad56e"
}
```

7.8 Error codes

The table below contains the error codes that the Pragmatic Play system will return in the response to Game History API calls.

Code	Description
0	Request was successfully processed.
1	Authentication failed. Incorrect secure login and secure password combination.
2	Request validation failed. Type mismatch for value.
4	Round not found. The requested roundId is not found in the system.
1000	Internal server error. Pragmatic Play will return this error code if their system has internal problem and cannot process the request.

VIII. Data feeds and reports

Data feeds let Casino Operator to download information about in-game money transactions for all game rounds played and jackpot winnings.

Data of the feeds related to large number of records (like transactions) will be requested using time point and sent to the Operator in CSV format.

For each API account, frequency of any particular "Data feeds and reports" method requests, should be not more than one call per minute.

Any "Data feeds and reports" method should be requested only with dataType parameter value, which is available for Operator. Available dataType values can be received from 2.1 GetCasinoGames method

Time points

Time point is a timestamp starting from the server time when data is requested for the first time. If the request is sent without a time point, the system returns a time point, which should be sent with the next request.

Whenever a request with the time point is sent, data from the moment of timepoint to the current moment will be transferred, along with the new time point. This new time point should be used for the next request, and so on.

Using such a mechanism, no data will be lost. If clients haven't received response from the Pragmatic Play with the new time point, they will be able to re-send the request with the same time point, until requested data is received.

Data can be loaded by small portions for periods of time not longer than 10 minutes.

Using timepoints for past periods

If Operator needs to get data for a longer period they can send a number of subsequent requests setting up a starting timepoint and then using timepoints that the Pragmatic Play system returns in the response.

For example, if Operator needs to get data for the last N minutes, they can calculate the start date and time of the period as $Start = ("now" - N \text{ minutes})$, convert it into timestamp and send request to the Transactions API using this timestamp as a time point. Server will return data for 10 minutes starting from the timestamp sent in the request, along with the new time point. Using timepoints returned by the server and sending requests one after another Operator can get as many portions of data as they need, to download data that are already in the past.

Data Feed URL

URL of the data feeds service will be provided by Pragmatic Play for the production and test environments and looks like:

https://{API service domain}/IntegrationService/v3/DataFeeds/transactions/

https://{API service domain}/IntegrationService/v3/DataFeeds/failedtransactions/

https://{API service domain}/IntegrationService/v3/JackpotFeeds/jackpots/

https://{API service domain}/IntegrationService/v3/JackpotFeeds/jackpots/winnings/

Data feeds service is securely protected, hence please be sure:

- player's browser (end) must NOT be used as initiator of requests to the service;
- proper IPs are supplied to Pragmatic Play for whitelisting

Data formats

Data will be returned in CSV or JSON format. For more details please see the chapter where the data feed is described.

All successful server responses in CSV format will contain a new time point that should be sent then by the client to the server in the next request. Format of the server response is as follows:

```
#timepoint = {server time in ms}  
{Data in standard csv format}
```

Client should get the first line from the response, parse it and save the time point somewhere for the next request. The rest of the response should be parsed as normal CSV stream.

8.1 Data types

Name	Type	Description
login	String(30)	API user name (secureLogin value) for authentication in the Pragmatic Play API.
password	String(30)	API password (secret key value) for authentication in the Pragmatic Play API.
timepoint	Long	Time point
dataType	String(3)	Type of portfolio, indicates which portfolio Data Feeds should be returned Available options: RNG - Main portfolio games (video slots, classic slots etc.) LC - Live Casino portfolio R2 - Fishing games portfolio VSB - Virtual sports betting portfolio Important! <i><u>In cases where dataType is included in request, Data Feeds for relevant portfolio will be returned only</u></i> <i><u>When there is no dataType in request, will be returned Data Feeds for RNG only</u></i>
playerID	Number(15)	Player unique identifier within Pragmatic Play system.
extPlayerID	String(100)	Player unique identifier within Casino Operator system.
gameID	String(20)	Symbolic unique identifier of the game provided by Pragmatic Play.
playSessionID	Number(15)	Id of the player's session for particular game (unique number of the game round).
parentSessionID	Number(15)	Id of the play session, in which the feature was triggered (unique number of the parent game round)
startDate	DateTime	Date and time when the game round started.
endDate	DateTime	Date and time when the game round ended.
status	String(1)	Status of the game round, transaction or jackpot
type	String(1)	Type of the game round or transaction
bet	Double(15,2)	Bet amount.
win	Double(15,2)	Win amount.
currency	String(3)	Currency of the transaction, 3 letter ISO code.
jackpot	Double(15,2)	Jackpot win amount.
platform	String(1)	The platform type (channel) on which the game was played. Possible values : U - Unknown W - WEB M - Mobile
bonusCode	String (100)	Id of the Free Round Bonus. Is applied for rounds, played with FRB
options	String(100)	List of settings
timestamp	DateTime	Date and time when the transaction is processed on the Pragmatic Play side.
referenceID	String(100)	Unique id for reference of this transaction on the Pragmatic Play side.
amount	Double(15,2)	Amount of the transaction.
betAmount	Double(15,2)	Amount of the bet.
winAmount	Double(15,2)	Amount of winning in current bet.

roundDetails	String(50)	Additional information about the current game round
jackpotID	Number(15)	Unique identifier of the Jackpot within Pragmatic Play system.
name	String(100)	Name of the Jackpot.
level	String(1)	Level of the Jackpot
games	String(255)	The list of the games participating in the Jackpot. It contains gameld (game symbols), comma separated.
dateTime	DateTime	Date and time when the Jackpot was won.
totalRounds	Integer	Total number of game rounds
totalFreeSpinRounds	Integer	Total number of free spin rounds
totalBet	Double(15,2)	Total bet amount.
totalWin	Double(15,2)	Total win amount.
totalJackpot	Double(15,2)	Total Jackpot win amount.
totalPromo	Double(15,2)	Total amount of wins in Prize Drop campaigns

8.2 Game Rounds

Using this data feed Casino Operator can load information about all game rounds, including not finished ones that were played during the period defined by time point, but not longer than 10 minutes.

There two options for getting game rounds data.

Request path: GET /DataFeeds/gamerounds/

Returns all game rounds including unfinished. Unfinished game rounds may appear in the feed twice – first when player makes a bet and then when player gets a winning.

Request path: GET /DataFeeds/gamerounds/finished/

Returns only game rounds that were finished within the time interval. Each game rounds appears in the data feed only once, after it is completed.

Data will be returned in CSV format.

Request parameters

Name	Description	Status
login	API user name (secureLogin value) for authentication in the Pragmatic Play API.	Required
password	API password (secret key value) for authentication in the Pragmatic Play API.	Required
timepoint	Time point starting from which data will be transferred. If time point is empty, the new time point will be returned. (Unix epoch time in milliseconds, for example : 1470926696715)	Optional
dataType	Type of portfolio, indicates which portfolio Data Feeds should be returned <u>In cases where dataType is included in request, Data Feeds for relevant portfolio will be returned only</u> <u>When there is no dataType in request, will be returned Data Feeds for RNG only</u>	Optional

Response parameters (CSV)

Name	Description	Status
playerID	Player unique identifier within Pragmatic Play system.	Required
extPlayerID	Player unique identifier within Casino Operator system.	Required
gameID	Symbolic unique identifier of the game provided by Pragmatic Play.	Required
playSessionID	Id of the player's session for particular game (unique number of the game round).	Required
parentSessionID	Id of the play session, in which the feature was triggered (unique number of the parent game round). This field will contain the same value as the playSessionID field for all game rounds except free spins. (Rows with free spins will contain the id of game round where they were triggered in the parentSessionID field and id of the free spin round in the playSessionID field)	Required
startDate	Date and time when the game round started. (Will be returned in yyyy-mm-dd HH:mm:ss format, for example: "2017-08-28 02:14:13")	Required
endDate	Date and time when the game round ended. Will be null if the game round is not finished yet. (Will be returned in yyyy-mm-dd HH:mm:ss format, for example: "2017-08-28 02:14:13")	Required

status	Status of the game round: I – In Progress (not finished yet) C – Completed	Required
type	Type of the game round: R – game round F – free spin triggered during the game round	Required
bet	Bet amount.	Required
win	Win amount.	Required
currency	Currency of the transaction, 3 letter ISO code.	Required
jackpot	Jackpot win amount.	Required
platform	The platform type (channel) on which the game was played. <i>The field is optional and is not handled by the Pragmatic Play by default. In case Casino Operator need this parameter to be applied, they should ask the Pragmatic Play's Technical Support for configuration.</i> Possible values : U - Unknown W - WEB M - Mobile	Optional
bonusCode	Id of the Free Round Bonus. Is applied for rounds, played with FRB <i>The field is optional and is not handled by the Pragmatic Play by default. In case Casino Operator need this parameter to be applied, they should ask the Pragmatic Play's Technical Support for configuration.</i>	Optional

Example of HTTP request:

```
GET /IntegrationService/v3/DataFeeds/gamerounds/?
login=username&password=testKey&timepoint=1482510713000 HTTP/1.1
Host: api.prerelease-env.biz
Cache-Control: no-cache
```

Example of response:

```
timepoint=1482510809258
playerID,extPlayerID,gameID,playSessionID,parentSessionID,startDate,endDate,status,type,bet,win,cur
rency,jackpot,bonusCode
6638030,bobtest,vs50aladdin,51126041,51126041,"2017-08-15 02:12:49","2017-08-15 02:18:14",C,R,
2.50,0.25,USD,0.00,some_bonus_code
6638030,bobtest,vs50aladdin,51126042,51126041,"2017-08-15 02:14:35","2017-08-15 02:14:40",C,F,
0.00,0.00,USD,0.00,some_bonus_code
6638030,bobtest,vs50aladdin,51126590,51126041,"2017-08-15 02:14:43","2017-08-15 02:14:46",C,F,
0.00,0.00,USD,0.00,some_bonus_code
6638030,bobtest,vs50aladdin,51127141,51126041,"2017-08-15 02:14:51","2017-08-15 02:14:55",C,F,
0.00,0.00,USD,0.00,some_bonus_code
```


8.3 In-game transactions

Request path: GET /DataFeeds/transactions/

Using this data feed Casino Operator can load money transactions for all game rounds that were played during the period defined by time point, but not longer than 10 minutes. Data will be returned in CSV format.

Request parameters

Name	Description	Status
login	API user name (secureLogin value) for authentication in the Pragmatic Play API.	Required
password	API password (secret key value) for authentication in the Pragmatic Play API.	Required
timepoint	Time point starting from which data will be transferred. If time point is empty, the new time point will be returned. <i>(Unix epoch time in milliseconds, for example : 1470926696715)</i>	Optional
dataType	Type of portfolio, indicates which portfolio Data Feeds should be returned <u>In cases where dataType is included in request, Data Feeds for relevant portfolio will be returned only</u> <u>When there is no dataType in request, will be returned Data Feeds for RNG only</u>	Optional
options	List of settings. By including it, operator can get additional information about transaction. Possible values are: <ul style="list-style-type: none">addTransactionStatus	Optional

Response parameters (CSV)

Name	Description	Status
playerID	Player unique identifier within Pragmatic Play system.	Required
extPlayerID	Player unique identifier within Casino Operator system.	Required
gameID	Symbolic unique identifier of the game provided by Pragmatic Play.	Required
playSessionID	Id of the player's session for particular game (unique number of the game round).	Required
timestamp	Date and time when the transaction is processed on the Pragmatic Play side. <i>(Unix epoch time in milliseconds, for example : 1470926696715)</i>	Required
referenceID	Unique id for reference of this transaction on the Pragmatic Play side.	Required
type	Type of the transaction: B – player made a bet W – player won L – Cancel bet transaction (for finalized rounds) R – refund transaction J – Player won Jackpot P – Win in promotional campaign	Required
amount	Amount of the transaction.	Required
currency	Currency of the transaction, 3 letter ISO code.	Required
status	Current status of transaction. Possible values are: S – success L – canceled R – refunded <i>This field is optional and will appear in response in cases where request contains <u>addTransactionStatus</u> in <u>options</u> list</i>	Optional

Example of HTTP request:

```
GET /IntegrationService/v3/DataFeeds/transactions/?  
login=username&password=testKey&timepoint=1482510713000&options=addTransactionStatus  
HTTP/1.1  
Host: api.prerelease-env.biz  
Cache-Control: no-cache
```

Example of response:

```
timepoint=1482510809258  
playerID,extPlayerID,gameID,playSessionID,timestamp,referenceID,type,amount,currency,status  
6638030,421,vs50aladdin,5111126041,1482510741000,585d5195f89c56f5ed95012e,B,2.50,USD,S  
6638030,421,vs50aladdin,5111126041,1482510742000,585d5196f89c56f5ed950159,W,0.50,USD,S  
6638030,421,vs50aladdin,5111126590,1482510745000,585d5199f89c56f5ed950344,B,2.50,USD,S  
6638030,421,vs50aladdin,5111127141,1482510748000,585d519cf89c56f5ed95054e,B,2.50,USD,S
```

8.4 Failed Transactions

Request path: GET /DataFeeds/failedtransactions/

Feed for getting Refund and BetResult/Result transactions which weren't processed even during asynchronous sending.

Request parameters

Name	Description	Status
login	API user name (secureLogin value) for authentication in the Pragmatic Play API.	Required
password	API password (secret key value) for authentication in the Pragmatic Play API.	Required
timepoint	Timepoint starting from which data will be transferred. If timepoint is empty, the new timepoint will be returned. (Unix epoch time in milliseconds, for example : 1470926696715)	Optional

Response parameters (CSV)

Name	Description	Status
playerID	Player unique identifier within Pragmatic Play system.	Required
extPlayerID	Player unique identifier within Casino Operator system.	Required
gameID	Symbolic unique identifier of the game within the Pragmatic Play system. <i>Example: vs25queenofgold, vs50aladdin, cs5triple8gold, bjma, rla, vpa</i> The list of available games may be obtained by Casino Operator by calling GetCasinoGames method of CasinoGameAPI HTTP service.	Required
playSessionID	Id of the player's session for particular game (unique number of the game round).	Required
referenceID	Unique id for reference of this transaction on the Pragmatic Play side.	Required
betAmount	Amount of the bet.	Required
winAmount	Amount of winning in current bet.	Required
timestamp	Date and time when the transaction is processed on the Pragmatic Play side. (Unix epoch time in milliseconds, for example : 1470926696715)	Required
roundDetails	Additional information about the current game round, comma separated. Example: "spin,bonus"	Required

Example of HTTP request:

```
GET /IntegrationService/v3/DataFeeds/failedtransactions/?  
login=username&password=testKey&timepoint=1482510713000 HTTP/1.1  
Host: api.prerelease-env.biz  
Cache-Control: no-cache
```

Example of response:

```
timepoint=1482510979767  
playerID,extPlayerID,referenceID,gameID,playSessionID,betAmount,winAmount,timestamp,roundDetails
```

8.5 Active Jackpots

Request path: GET /JackpotFeeds/jackpots/

Active Jackpots feed provides information about open jackpots per casino brand. Operator will get in the response a list of the jackpots in JSON format.

Request parameters

Name	Description	Status
login	API user name (secureLogin value) for authentication in the Pragmatic Play API.	Required
password	API password (secret key value) for authentication in the Pragmatic Play API.	Required
currency	ISO-4217 code currency. Currency for JP amounts returned in feed. If missed, then amounts are returned in USD.	Optional

Response parameters (JSON)

Name	Description	Status
jackpotID	Unique identifier of the Jackpot within Pragmatic Play system.	Required
name	Name of the Jackpot.	Required
level	Level of the Jackpot: N – Network jackpot, O – Jackpot for particular Operation B – Jackpot for particular casino Brand	Required
amount	Jackpot fund for the moment of request, in USD by default. Or values can be returned in the requested currency.	Required
games	The list of the games participating in the Jackpot. It contains gameld (game symbols), comma separated. <i>Example: vs25safari_jp,vs30catz_jp,vs7monkeys_jp</i>	Required
status	Current status of the Jackpot. Possible values : A - Active S - Shutdowned	Required

Example of HTTP request:

```
GET /IntegrationService/v3/JackpotFeeds/jackpots/?login=username&password=testKey HTTP/1.1
Host: api.prerelease-env.biz
Cache-Control: no-cache
```

Example of JSON response:

```
{
  "jackpots": [
    {
      "jackpotID": 569,
      "name": "test_pp_jp1",
      "level": "B",
      "amount": 1,
      "games": "vs25safari_jp",
      "status": "A"
    }
  ],
  "error": "0",
  "description": "OK"
}
```

8.6 Jackpot Winnings

Request path: GET /JackpotFeeds/jackpots/winnings/

The Jackpot Winnings data feed provides information about last jackpots winnings. The feed returns in the response a list of last 10 winnings of specified jackpot, in JSON format.

Request parameters

Name	Description	Status
login	API user name (secureLogin value) for authentication in the Pragmatic Play API.	Required
password	API password (secret key value) for authentication in the Pragmatic Play API.	Required
jackpotID	Unique identifier of the Jackpot within the Pragmatic Play system.	Required

Response parameters (JSON)

Name	Description	Status
jackpotID	Unique identifier of the Jackpot within Pragmatic Play system.	Required
dateTime	Date and time when the Jackpot was won.	Required
amount	Amount of Jackpot that was won, \$USD	Required
gameID	Symbolic unique identifier of the game where Jackpot was won.	Required

Example of HTTP request:

```
GET
/IntegrationService/v3/JackpotFeeds/jackpots/winnings/?login=username&password=testKey&jackpotID=568 HTTP/1.1
Host: api.prerelease-env.biz
Cache-Control: no-cache
```

Example of JSON response:

```
{
  "winnings": [
    {
      "jackpotID": 568,
      "dateTime": "2016-12-23 06:31:54",
      "amount": 55,
      "gameID": "vs30catz_jp"
    }
  ],
  "error": "0",
  "description": "OK"
}
```

8.7 Daily Totals

Request path: GET /DataFeeds/totals/daily/

Daily Totals API provides aggregated data grouped by currency for every day from a certain period. Operator can use this method for cross checking their data. Only completed game rounds are included in the response. If a game round was started in one day and finished in another day, its results will be included in the daily totals for the day it was finished. Data will be returned in JSON format.

Request parameters

Name	Description	Status
login	API user name (secureLogin value) for authentication in the Pragmatic Play API.	Required
password	API password (secret key value) for authentication in the Pragmatic Play API.	Required
startDate	Start date and time of the period for which the data should be loaded	Required
endDate	End date and time of the period for which the data should be loaded	Required
dataType	Type of portfolio, indicates which portfolio Data Feeds should be returned <i><u>In cases where dataType is included in request, Data Feeds for relevant portfolio will be returned only</u></i> <i><u>When there is no dataType in request, will be returned Data Feeds for RNG only</u></i>	Optional
options	List of settings. By including it, operator can get additional totals. Possible values are: <ul style="list-style-type: none">• <u>addTotalPromo</u> – for Prize Drop wins	Optional

Response parameters (JSON)

Name	Description	Status
totalRounds	Total number of game rounds completed during the reporting period, including free spin rounds number	Required
totalFreeSpinRounds	Total number of free spin rounds completed during reporting period	Required
totalBet	Total bet amount.	Required
totalWin	Total win amount.	Required
totalJackpot	Total Jackpot win amount.	Required
totalPromo	Total amount of wins in Prize Drop campaigns. <i>This field is optional and will appear in response in cases where request contains <u>addTotalPromo</u> in <u>options</u> list</i>	Optional
currency	Currency, 3 letter ISO code.	Required

Example of HTTP request:

```
GET /IntegrationService/v3/DataFeeds/totals/daily/?login=username&password=testKey&startDate=2018-05-21 00:00:00&endDate=2018-05-22&options=addTotalPromo
23:59:59 HTTP/1.1
Host: api.prerelease-env.biz
Cache-Control: no-cache
```

Example of JSON response:

```
{ "error": "0",
  "description": "OK",
  "data": [
    {
      "totalRounds": 48,
      "totalFreeSpinRounds": 10,
      "totalBet": 48,
      "totalWin": 92.68,
      "totalJackpot": 0,
      "totalPromo": 0,
      "currency": "RMB"
    }
  ]
}
```

8.8 Player's Incomplete Rounds

Request path: GET /DataFeeds/gamerounds/incomplete/

This report returns all the game rounds that were started by a certain player but not completed yet.

Data will be returned in JSON format.

Request parameters

Name	Description	Status
login	API user name (secureLogin value) for authentication in the Pragmatic Play API.	Required
password	API password (secret key value) for authentication in the Pragmatic Play API.	Required
playerId	Id of the player within the Casino Operator's system.	Required
dataType	Type of portfolio, indicates which portfolio Data Feeds should be returned <u>In cases where dataType is included in request, Data Feeds for relevant portfolio will be returned only</u> <u>When there is no dataType in request, will be returned Data Feeds for RNG only</u>	Optional

Response parameters (JSON)

Name	Description	Status
playerId	Id of the player within the Casino Operator's system.	Required
gameId	Id of the game.	Required
playSessionID	Id of the player's session for particular game (unique number of the game round).	Required
betAmount	Bet amount.	Required

Example of HTTP request:

```
GET /IntegrationService/v3/DataFeeds/gamerounds/incomplete/?  
login=username&password=testKey&playerId= bobtest HTTP/1.1  
Host: api.prerelease-env.biz  
Cache-Control: no-cache
```

Example of JSON response:

```
{  
  "error": "0",  
  "description": "OK",  
  "playerId": "bobtest",  
  "data": [  
    {  
      "gameId": "vs50aladdin",  
      "playSessionID": "2352191424",  
      "betAmount": 10.00  
    },  
    {  
      "gameId": "vs20b1",  
      "playSessionID": "23521914534",  
      "betAmount": 20.00  
    },  
    {  
      "gameId": "vs20cm",  
      "playSessionID": "2352191665",  
      "betAmount": 30.00  
    }  
  ]  
}
```


IX. Tournament API

Tournament API allows operators to set up an automated procedures related to Pragmatic tournaments.

9.1 Tournament Winners

Request path: GET /tournaments/winners/

This API method returns winners for all the tournaments that were closed during requested period of time.

Data will be returned in JSON format.

Request parameters

Name	Type	Description	Status
login	String(30)	API user name (secureLogin value) for authentication in the Pragmatic Play API.	Required
password	String(30)	API password (secret key value) for authentication in the Pragmatic Play API.	Required
startDate	Date	Start date of the period for which the data should be loaded. <i>Date and time in the format yyyy-MM-dd HH:mm:ss.</i>	Required
endDate	Date	End date of the period for which the data should be loaded. <i>Date and time in the format yyyy-MM-dd HH:mm:ss.</i>	Required

Response parameters (JSON)

Name	Type	Description
tournaments	List	List of tournaments that were closed during requested period. See below Tournament type description of the objects in the list
error	String	Error code
description	String	Description of the error for troubleshooting

Type: Tournament

Name	Type	Description
tournamentID	Number(15)	Id of the tournament
name	String	Name of the tournament
dateOpened	Date	Date and time when the tournament started. <i>Date and time in the format yyyy-MM-dd HH:mm:ss.</i>
dateClosed	Date	Date and time when the tournament was finished. <i>Date and time in the format yyyy-MM-dd HH:mm:ss.</i>
winners	List	Tournament winners – the list of top players that were awarded with tournament prizes. See below “Tournament Winner” type description of the objects in the list

Type: Tournament Winner

Name	Type	Description
playerID	String	Id of the player within the operator’s system.
tournamentPlayerID	Number(15)	Player’s id within the tournament

position	Number(5)	Rank of the player in the tournament
score	Number(15)	Score of the player
prizeGift	String	Gift, which the player will receive as a prize. This field is optional and will appear only if prizeAmount, prizeCoins, prizeCurrency fields are not present in the response.
prizeAmount	Number(15,2)	Prize amount that the player is awarded with. <i>The field is optional and will appear only if there is no prizeGift field in the response. Please note, that prizeAmount, prizeCoins, prizeCurrency should appear together.</i>
prizeCoins	Number(15)	Prize amount in coins. <i>The field is optional and will appear only if there is no prizeGift field in the response. Please note, that prizeAmount, prizeCoins, prizeCurrency should appear together.</i>
prizeCurrency	String	Currency code for prize amount. <i>The field is optional and will appear only if there is no prizeGift field in the response. Please note, that prizeAmount, prizeCoins, prizeCurrency should appear together.</i>

Example of HTTP request:

```
GET
/IntegrationService/v3/http/tournaments/winners/?login=username&password=testKey&startDate=
2018-01-20 11:00:00&endDate=2018-01-20 12:00:00 HTTP/1.1
Host: api.prerelease-env.biz
Cache-Control: no-cache
```

Example of JSON response:

```
{
  "error" : 0,
  "description" : "OK",
  "tournaments" : [ {
    "tournamentID" : 10,
    "name" : "Super tournament",
    "dateOpened" : 2018-01-18 10:00:00,
    "dateClosed" : 2018-01-20 11:30:00,
    "winners" : [ {
      "playerID" : "extid-0",
      "tournamentPlayerID" : 20,
      "position" : 1,
      "score" : 100000,
      "prizeAmount" : 1000.00,
      "prizeCoins" : 20.0,
      "prizeCurrency" : "USD"
    },
    {
      "playerID" : "extid-3",
      "tournamentPlayerID" : 24,
      "position" : 2,
      "score" : 99667,
      "prizeAmount" : 500.0,
      "prizeCoins" : 10.0,
      "prizeCurrency" : "USD"
    },
    {
      "playerID" : "extid-6",
      "tournamentPlayerID" : 28,
      "position" : 3,
      "score" : 99334,
      "prizeAmount" : 250.0,
      "prizeCoins" : 5.0,
      "prizeCurrency" : "USD"
    }
  ]
} ]
}
```

X. Business Statistics API

Statistics API is used for receiving business statistics reports from Pragmatic Play about play activity, based on needed options.

URL of the HTTP service will be provided by Pragmatic Play for the production and test environments and looks like:
`https://{API service domain}/IntegrationService/v3/StatisticsAPI/players/daily/totals/`

Integration simple HTTP Service is securely protected, hence please be sure:

- player's browser (end) must NOT be used as initiator of requests to the service;
- proper IPs are supplied to Pragmatic Play for whitelisting

Hash calculation

Hash code is calculated with following formula:

From request GET parameters, all parameters are taken (except hash) and appending to string:

1. Sort all parameter by keys in alphabetical order.
2. Append them (if the value is not null or empty) in `key1=value1&key2=value2`.
3. Append secret key, e.g.: `key1=value1&key2=value2SECRET`.
4. Calculate the hash by using MD5.

10.1 Players daily report

Players Daily report provides aggregated data grouped by player and optionally by game for. Player statistics is accumulated by day based on Pragmatic Play server time zone. Operator can request data for a certain day only.

If operator requests data for the current day, the last aggregated data will be returned, but in order to receive statistics for the whole day, this report should be requested after day is finished.

Operator can use this method for cross checking their data. Only completed game rounds are included in the response. If a game round was started in one day and finished in another day, its results will be included in the daily totals for the day it was finished.

Request path: `GET /StatisticsAPI/players/daily/totals/`

Request parameters

Name	Type	Description	Status
secureLogin	String	User name for authentication in the Report API service.	Required
hash	String	Hash code of the request.	Required
date	String	Date for which the data should be loaded (format YYYY-MM-DD)	Required
options	String	List of settings. By including it, operator can get an additional totals or group by additional parameters. Possible values are: <i>groupByGame</i> – if it is required to receive data per player per game	Optional

Response parameters (CSV)

Name	Type	Description
playerID	String	Id of the player within the Casino Operator's system
gameID	String	Symbolic unique identifier of the game within the Pragmatic Play system. <i>Example: vs25queenofgold, vs50aladdin, cs5triple8gold, bjma, rla, vpa</i> In case options parameter is set as groupByGame
totalRounds	Number	Total number of game rounds completed during the reporting period, does not include free rounds received as promo.
totalFreeSpinRounds	Number	Total number of free spin rounds (in-game feature) completed during reporting period.
totalBet	Double(15,2)	Total bets made by player during reporting period. The values are in player's currency.
totalWin	Double(15,2)	Total wins won by player during reporting period. The values are in player's currency.
currency	String	Player's currency, 3 letter ISO 4217 code. Example: "USD"

Example of HTTP request:

```
GET /IntegrationService/v3/StatisticsAPI/players/daily/totals/?
secureLogin=login&hash=34642b49d19990ff97a13ed6384554e4&date=2020-03-04&options=groupByGame
HTTP/1.1
Host: api.prerelease-env.biz
Cache-Control: no-cache
```

Example of response:

```
playerID,gameID,totalRounds,totalFreeSpinRounds,totalBet,totalWin,currency
player1 ,vs10bookoftut,5,0,10.00,0.00,USD
player2,vs25mmouse,11,0,22.00,1.60,USD
```

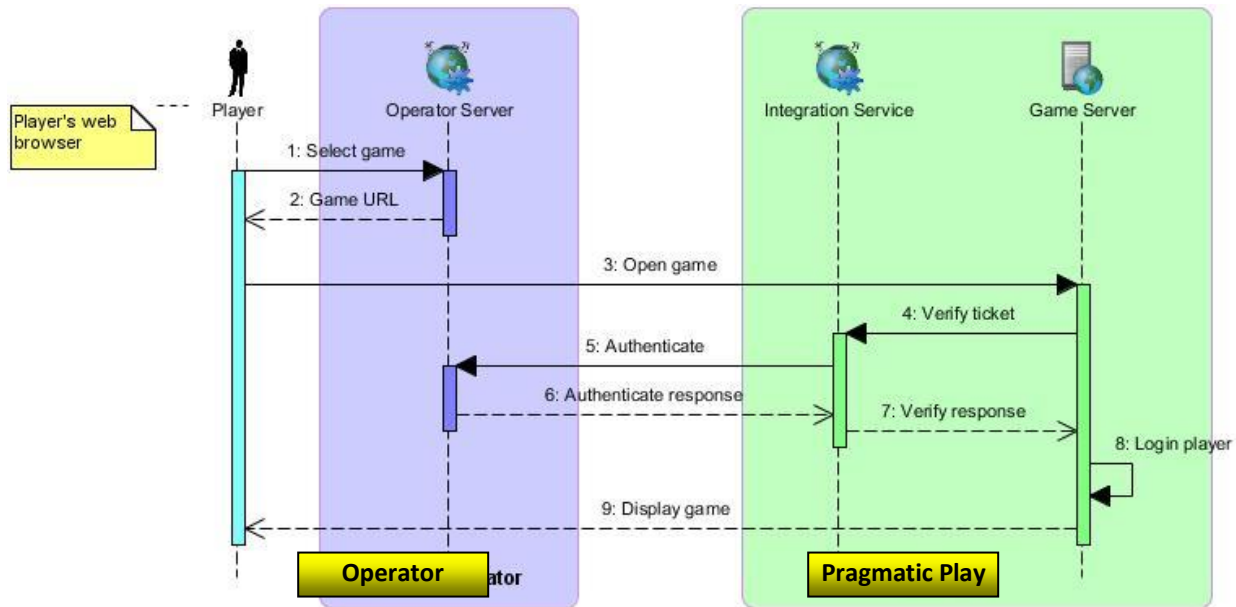
10.2 Error codes

Below table contains the error HTTP statuses that should be returned, in case of issues on request processing

HTTP status	Description
400	Bad request. Is returned in cases when some of required parameters are not present, or empty
401	Unauthorized - Incorrect secure login or calculated hash
500	Internal error. Is returned when there are some problems, or data cannot be returned because of invalid date in request

XI. Diagrams

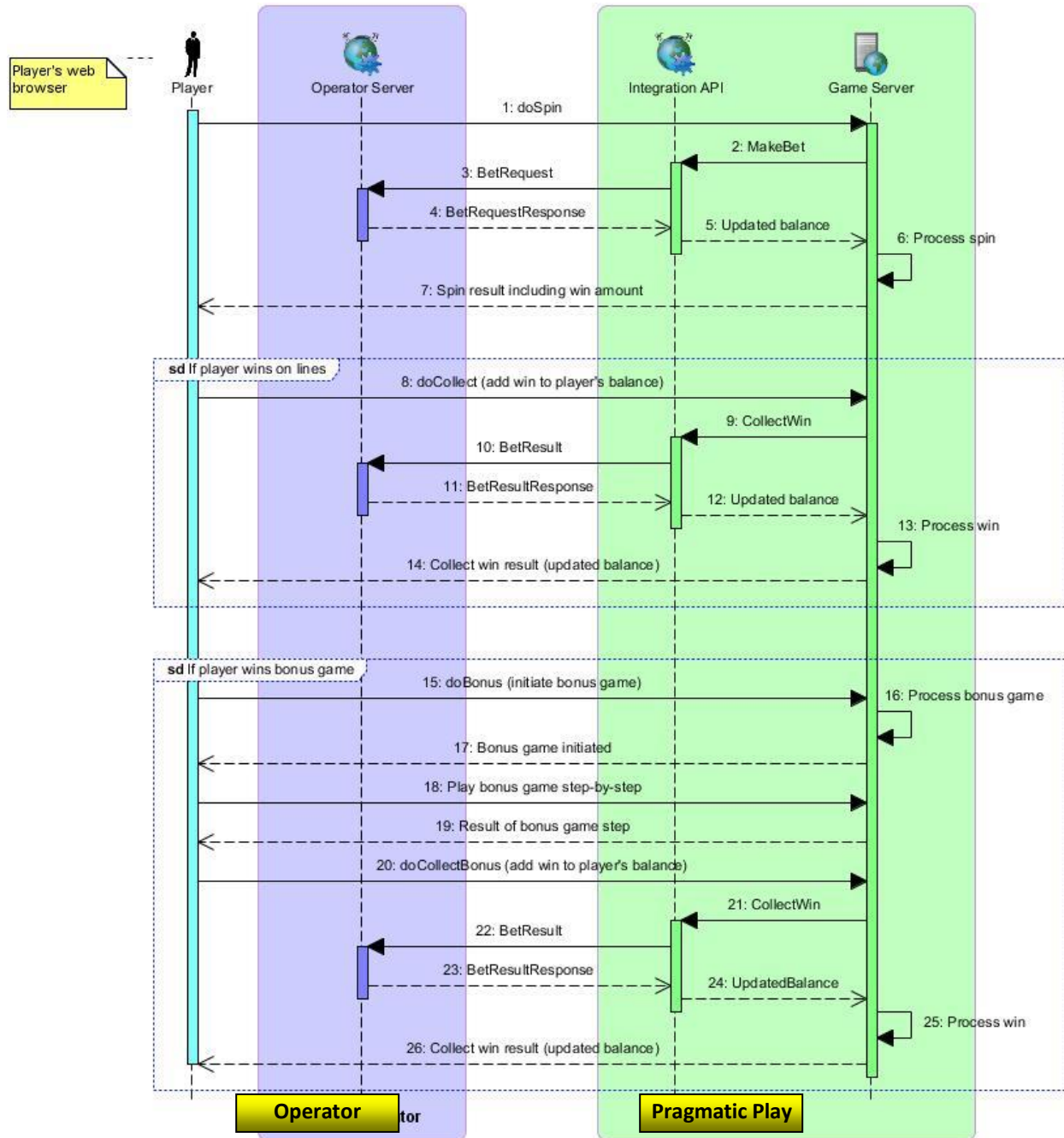
11.1 Flow diagram of opening casino game for Seamless Wallet API



Flow details

1. Player selects a game, which is served by Pragmatic Play. Specific one-time password is created on Operator side. This password should uniquely identify the player within Operator system.
2. Operator, with the help of JS library provided by Pragmatic Play, checks if selected game can be launched on the player's OS and return URL of the game to the player browser. If the game cannot be launched then appropriate message will be shown to the player.
3. URL of the game is loading in player browser for redirecting player to Pragmatic Play game server.
4. Game Server verifies one-time password received with URL.
5. Pragmatic Play Integration Service calls Operator server to authenticate player using received token.
6. Operator authenticates the player. If player is authenticated Operator returns player id within Operator system and player current balance.
7. Game Server receives a result of player authentication. If authentication is not successful a player will receive an error message.
8. Game Server finds player id in the database or creates a new player account
9. The game with all required data is opened for player.

11.2 Flow diagram of playing casino game Seamless Wallet API



Flow details

The game flow is shown in the example of the slot.

1. Player clicks Spin button on the game screen, and game client sends doSpin request to the Pragmatic Play Game server.
2. Game server calls Pragmatic Play Integration service to withdraw money from the player balance.
3. Integration service sends Bet request to the Operator server. If the request cannot be handled by Operator due to connection problem, Pragmatic Play server will repeat the request 3 times. After that Refund request will be sent to the Operator server for cancelling bet until response from Operator server is received.
4. Operator verifies that player has enough funds in the wallet to make a bet, subtracts requested amount from the balance and returns updated player balance in the response. If there is not enough money in the wallet, Operator returns an error.
5. Pragmatic Play Integration service returns updated balance to the game server.
6. Game server processes spin - gets new combination on the reels, checks combinations that trigger free spins or bonus game feature, calculates win amount, etc.
7. Game server returns response to the game client. Response may contain updated reels, information about winning lines, triggered features and win amount.
8. If player has a win on lines game client sends doCollect request to the Game server to collect player's winnings.
9. Game server calls Pragmatic Play Integration service to add win amount to the player balance.
10. Integration service calls Operator server sending BetResult request. If request cannot be handled by Operator due to connection problem, Pragmatic Play server will repeat the request until it receives a response from Operator server.
11. Operator adds money to player's wallet and returns updated player balance.
12. Pragmatic Play Integration service returns updated balance to the Game server.
13. Game server collects player's win.
14. Game server returns response to the game client containing updated player balance.
15. If bonus game feature is triggered game client sends doBonus request to the Game server.
16. Game server starts bonus game feature.
17. Game server returns response with all information that is necessary for the game client to play bonus game.
18. Game client sends a request to the game server on every step of the bonus game.
19. Game server handles request and returns response with the result of current step.
20. When bonus game is over game client sends doCollectBonus request to the Game server.
21. Game server calls Pragmatic Play Integration service to add bonus game win amount to the player balance.
22. Integration service calls Operator server, sending BetResult request. If request cannot be handled by Operator due to connection problem, Pragmatic Play server will repeat request until it receives a response from Operator server.
23. Operator adds money to player's wallet and returns updated player balance.
24. Pragmatic Play Integration service returns updated balance to the Game server.
25. Game server processes bonus game win.
26. Game server returns response to the game client containing updated player balance.

XII. Reconciliation

12.1 Reconciliation process

If a request times out because of internet connection problems, or contains relevant error code (Error codes), then the Pragmatic Play system will follow a process described below, to reconcile the action.

Bet transaction

The Bet request will be retried 2 (two) times with a delay of 5 seconds between attempts.

If a request is still not processed after the second try, Pragmatic Play will initiate the Refund request to cancel this single bet on the Operator side.

Refund transaction

Refund request will be added to a Transaction queue and an error message will be displayed to the player on the game interface. The game client should be closed. The reconciliation mechanism starts from there.

Operator will accept Refund requests that run independently of the game session in order to ensure the two systems are kept in sync.

Such a request can be sent if Pragmatic Play has an unknown response on the Refund request on its side because of a timeout or relevant error code (so it will try to run it in the background until the Operator response will not reach Pragmatic Play).

Operator will return a successful response on the Refund request even if the transaction has been canceled in the past and does not need to be canceled again or even if Bet has not reached Operator system because of timeouts.

Winning transactions

The BetResult/BonusWin/JackpotWin requests will be retried 2 (two) times with a delay of 5 seconds between attempts.

If a request is still not processed after the second try then it will be added to the Transaction queue for reconciliation. Game round result will be displayed to the player on the game interface.

BetResult/BonusWin/JackpotWin requests will be added to the Transaction queue and sent independently of the game session, according to the Reconciliation mechanism.

End round

The EndRound request will be added to the Transaction queue once the game server determines a game round is over. The reconciliation mechanism starts from there.

Operator will accept EndRound requests that run independently of the game session in order to ensure the two systems are kept in sync. Repeated EndRound request can be sent only if the Pragmatic Play system has an unknown status on its side because of a timeout or special error code on the EndRound call, so it will try to run it in the background until the Operator response will not reach Pragmatic Play.

12.2 Reconciliation mechanism

The reconciliation mechanism ensures the two systems (Game Platform and Casino Operator) are kept in sync.

From the Pragmatic Play system perspective, the Transaction queue is a table within the database that records any Refund, BetResult/BonusWin/JackpotWin and EndRound requests that timed out.

The Pragmatic Play system put in place an automated process (cron job) that reads the Transaction queue table from the database and runs the requests within the queue.

After 24 hours, the requests that are still unsuccessful will be marked for manual reconciliation.

XIII. Reality Check

13.1 Server side implementation

Reality Check Warning

One of the regulatory requirement is to notify the player with a certain frequency regarding the actual amount of time the player spends in the game. In UK regulation the user set the notification frequency during the login via the portal. In the next money transaction call (bet or win) after the reality check interval expiration will result in an appropriate reality check error. The error should be translated to an appropriate warning popup displayed to the player.

Reality check error

Error code	Description	Additional data
210	Reality check warning	<ul style="list-style-type: none">rcRegulation – Regulator. Possible values are: UK, MaltarcInterval – Reality check interval in minutesrcTimeInSession – Total session time in minutesrcNetGain – Status of player's profit/loss over the session time

Example of reality check error response:

```
{
  "error": 210,
  "description": "Reality check warning",
  "details": {
    "rcRegulation": "Malta",
    "rcInterval": 20,
    "rcTimeInSession": 60,
    "rcNetGain": 23.05
  }
}
```

Reality Check Response

Request path: POST /realityCheck/response/

Once user have received reality check error, they have two choices – continue to play or stop playing and logout. Pragmatic Play will send the Reality Check Response request, in order to notify the Operator about player's choice. Operator should resume the player's session, reset the reality check counter and continue handling money transactions.

If player have chosen to stop playing and there is a winning requests that were refused by Operator due to the reality check warning, those requests must be properly processed on both sides to allow player to get their winnings: Pragmatic Play will re-send winning requests and Operator must add money to player's balance.

Request parameters

Name	Description	Status
providerId	Pragmatic Play identifier for authentication in Casino Operator's system. May be provided by Casino Operator. Otherwise default value is used.	Required
playerId	Identifier of the player within the Casino Operator system.	Required
userAction	Represent the user response to reality check dialog. Can be one of the 2 options below: <ul style="list-style-type: none">○ Reset – Continue to play and reset the reality check counter○ Stop – Stop playing. Please note, in such case all active game windows should be terminated	Required
hash	Hash code of the request.	Required

Response parameters

Name	Description
error	Error code.
description	Description of the error for troubleshooting.

Example of HTTP request:

```
POST /realityCheck/response/ HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

playerId=6&userAction=Reset&providerId=PragmaticPlay&hash=53ff1b78529a7b97fca9026d9c2dc788
```

Example of JSON response:

```
{
  "error": "0",
  "description": "Success"
}
```

13.2 Client side implementation

The game client should calculate a time the player has spent in the game and display a reality check message every X minutes, which will inform the player how long they are playing the game. To activate reality check feature in the game client an Operator should provide the Pragmatic Play game launcher with the following parameters:

- rci – the reality check interval, in minutes. This is the interval for reality check messages to be shown to the player during game play
- rce – the reality check elapsed time, in minutes. This is optional parameter, which indicates already elapsed minutes on another games. If this parameter is sent by Operator, the first time RC message will appear in rci – rce minutes after game opening
- rcHistoryUrl – a link to the player's game history. This is optional parameter. If Operator send the rcHistoryUrl parameter, game client will add the "History" button to the reality check message. There are two options here:
 - Operator may provide a URL to the page on their side. In this case rcHistoryUrl should contain a link to Operator's page. For example: rcHistoryUrl=http://casinosite.com/history/
 - Game client can show inbuilt game history, provided by Pragmatic Play. In this case Operator should send the predefined value: rcHistoryUrl=ingame
- rcCloseUrl – a link to the page on the Operator's website, to which the player will be redirected if they choose to close the game. This parameter is optional. The "Close" button appears on the reality check message only if Operator sends the URL

The "Continue" button always appears on the reality check message. This button removes the reality check message and resumes the game screen.

XIV. Custom messages

Seamless Wallet API responses can optionally contain one or more messages that will be shown in the player's game client. This optional feature initially is designed for UKGC CMA support, but operators can also use it for their needs.

Upon receiving a message, game client will display information to the player in order to inform him or her about a specific event, and optionally request a user response.

All text that is sent inside a message must be translated to the player's language.

14.1 Custom messages syntax

A custom message is an optional "message" element added to API responses. There can be zero or more "message" elements per response:

```
{...
  "messageTriggers": [
    {
      "title": "Starting to use bonus funds",
      "text": "You are starting play with bonus funds.\nFrom this point it possible that won funds may have to be\nre-bet many times before they can be withdrawn from your account.\nPlease click Bonus Details for further details.",
      "nonIntrusive": false,
      "options": [
        {
          "action": "continue",
          "label": "OK"
        },
        {
          "action": "link",
          "label": "Bonus Details",
          "linkType": "redirect",
          "url": " https://www.casino.com/bonus/details",
        }
      ]
    }
  ]
}
```

Each item in the "messageTriggers" array may contain elements `title`, `text`, `nonIntrusive` and `options`.

The `title` element contains the title of the message box, which must be in one short line of no more than 60 characters.

The `text` specifies the message box text and may contain multiple lines. When specifying a forced line break the symbol used is `\n`. The max length of the text is 300 characters.

If `nonIntrusive` is specified and true, the message can be shown in a non-modal, non-intrusive fashion. Default value: false.

The `options` array can contain one or more elements. Each option always has `action` and `label` elements and may contain optional `url` element for redirect or Ajax call and a `linkType` specifying how the request should be sent to the URL if it is specified. Options and link types are specified below.

The options order specifies the order in which the options shall appear in the message box.

14.2 Action types

Each `option` element inside a message has a type that specifies the expected behavior when that option is selected. These are the possible values:

Type	Use case	Description
quit	Quit the game	Close the game. Acts the same as pressing Home button.
continue	Continue the game	Close the message box and resume the game.
history	Game history/Details	Close the message box and open the history URL if it was specified during game launch. If history URL was not specified please use the "link" type.
link		Close the message box and open or send Ajax request to the URL indicated in the "action" parameter.

14.3 Link types

The action element inside an option specifies the expected behavior when that option is selected. The default value is redirect. These are the possible values:

Type	Use case	Description
redirect	Open a page	Redirects the player to the URL specified for the "link" option.
open	Open a new tab	Opens a new/blank tab in the browser, with the link specified for the action.
ajax	Notify and resume	An Ajax request is sent to the URL and immediately let the player continue with the game flow.
ajaxResponse	Notify and wait	An Ajax request is sent to the URL and wait until "successful json response" is received. Only then continue with the game flow.
notify	Notify and resume	A JavaScript event should be sent to parent window as <code>postMessage()</code> . For example, if the game is opened in <code>iframe</code> and player wants to go back to the lobby JS notification to the parent page should be used instead of <code>redirect</code> , which will cause opening the lobby page inside <code>iframe</code> .

14.4 Custom messages API

Response parameters

Name	Type	Description	Status
messageTriggers	Array	Messages that will be displayed in the player's game client.	Optional
message	Object		
title	String (60)	Title of the message box.	Optional
text	String (300)	Text to be shown in the message box.	Required
nonIntrusive	Boolean	If specified and true, the message can be shown in a non-modal, non-intrusive fashion. Default value: false.	Optional
options	Array	List of options, should specify buttons shown in the message box. At least one option should be defined. In the simplest case, it will be the OK button, which closes the box.	Optional
option	Object		
action	String (10)	Type of the option. Available types are: quit, continue, history, link. Detailed description of each action can be found in Action types section above.	Required
label	String (20)	Text on the button in the message box to be shown to the player.	Required
linkType	String (10)	Type of the link. Relevant only for actions of "link" type. Available values are: redirect, ajax, ajaxResponse, notify. Detailed description of each link type can be found in Link types section above.	Optional
url	String (100)	URL of the link. Relevant only for actions of "link" type.	Optional

XV. Regulated Markets

This section represents integration API and other useful materials important for certain regulated markets.

15.1 Responsible gaming - Sweden

For Sweden there are four items that needs additional attention in order to comply with Swedish regulations.

Login time

Players should at all-time be able to see login time, this includes the mobile client as well. For this purpose, PragmaticPlay takes an addition game launch parameter:

elapsedTime - Current elapsed session time, the value is in seconds. If present the elapsed time will be displayed in the game client.

Links and logos to self-assessment tests

Player should also at all-time be able to see three logos with links at the top of the client in a locked field. PragmaticPlay have extended our game launcher to take three additional launch parameters.

- **pauseplayUrl** – encoded url to the “Spelpaus”
- **selftestUrl** – encoded url to the “Självtest”
- **playlimitUrl** – encoded url to the “Spelgränser”

Set player's country

Operator needs to set jurisdiction on PragmaticPlay authenticate call, the jurisdiction parameter is used for reporting and Game Client behavior. For Sweden the jurisdiction is “SE”.

Displaying of totals

To display in game interface total amounts of Bet, Win or Loss during player's game session, operator should use section **XIII Custom messages** of current document.

15.2 Checksums API

Using this method Casino Operator will get the list containing all critical files with their checksums to be verified and sent to the Regulator.

All responses should be in JSON format.

URL of the API for regulated markets will be provided by Pragmatic Play for the production and test environments and looks like:

<https://{API service domain}/IntegrationService/v3/http/RegulationAPI/>

All HTTP services are securely protected, hence please be sure:

- player's browser (end) must NOT be used as initiator of requests to the service;
- proper IPs are supplied to Pragmatic Play for whitelisting

Request path: POST /checksums/

Request parameters

Name	Description	Status
secureLogin	User name for authentication in the Casino Game API service	Required
jurisdiction	One or more jurisdiction codes (delimited with comma) to filter the result. May be omitted to get all available jurisdictions	Optional
hash	Hash code of the request.	Required

Response parameters

Name	Type	Description
checksums	List	The list of filenames and checksums for every critical file.
error	String(10)	0 - if the request was successfully processed or error code in other case.
description	String(250)	Description of the error for troubleshooting purposes.

Example of HTTP request:

POST /IntegrationService/v3/http/RegulationAPI/checksums/ HTTP/1.1

Host: api.prerelease-env.biz

Content-Type: application/x-www-form-urlencoded

secureLogin=ext_test1&hash=3b4d0b1b84bf4eaa694acb0657a922f3&jurisdiction=IT,LV

Example of JSON response:

```
{
  "checksums": [
    {
      "files": [
        {
          "checksum": "3aa16f6a16adadd8065aa35f4a5ac9877c4408e6",
          "fileName": "vs50aladdin_cv2-88.01_.hs",
          "gameId": "vs50aladdin"},
        {
          "checksum": "48e923a19fa4460a8b0b589019813f08be460991",
          "fileName": "vs50aladdin_cv2-96.53_.hs",
          "gameId": "vs50aladdin"},
        ...
      ],
      "jurisdiction": "LV"
    },
    {
      "files": [
        {
          "checksum": "c39c25ad9a1128f3bf79bd578fc579521d9c5059",
          "fileName": "dh-gs2c-random-1.0.39189.jar",
          "gameId": "vs243lions"},
        {
          "checksum": "b33e33da96c37c0eb2d02b3fbf439685fcda876c",
          "fileName": "dh-rng-1.0.0.jar",
          "gameId": "vs243lions"},
        ...
      ],
      "jurisdiction": "IT"
    }
  ],
  "description": "OK",
  "error": "0"
}
```


15.3 Portuguese Regulated Market - Slot Machine Reporting

This is a simple API for Operators to connect to the Pragmatic Play gaming system. API is an HTTP listener, which listens POST requests coming to an URL with the request mappings below.

URL of the HTTP service will be provided by Pragmatic Play for the production and test environments and looks like:
https://{API service domain}/RegulationService/

The service is securely protected, hence please be sure:

- player's browser (end) must NOT be used as initiator of requests to the service;
- proper IPs are supplied to Pragmatic Play for whitelisting

Introduction

This document describes the API, which Pragmatic Play has created for Operators in Portugal, for slot machines reporting according to SRIJ requirements providing the way of getting the string needed to fill the smResult field present in the AJOG files.

API Architecture

The communication interface between the Operator and the Pragmatic Play is performed by server-to-server requests via HTTPS (using POST method). Responses are sent as JSON. The structure and the elements information are defined in this section.

Requests Authentication

All requests that are sent to the Pragmatic Play API should be authenticated. Each request contains Secure Login as a request parameter and a Password as a part of the hash code calculated based on the request parameters. Pragmatic Play will supply the Operator with dedicated credentials.

Error Handling

If the ErrorCode parameter equals to 0 or not exist, then the response was processed successfully with no error. In Case the ErrorCode > 0, it means that the response was processed with an error.

Slot Machine Reporting API

Hash calculation

Hash code is calculated with the following formula:

From request post parameters, all parameters are taken (except hash) and appending to string:

1. Sort all parameter by keys in alphabetical order.
2. Append them (if the value is not null or empty) in key1=value1&key2=value2.
3. Append secret key, e.g.: key1=value1&key2=value2SECRET.
4. Calculate the hash by using MD5.
5. Compare with hash parameter. In the case of failure Casino Operator should send the error code 1.

Credentials for connecting to Pragmatic Play service, for integration and production environments, will be provided by our Integration team by Operator's request.

API specification

Request path: POST /ReportingAPI/Portugal/smResult/

Using this method Operator can get a details of slot machine results for particular game round, which then can be sent to the Regulator in the smResult field.

Important: Only completed game rounds that have been finished by players are allowed. The smResult report cannot be generated for incomplete game rounds.

Request parameters

Name	Type	Description	Status
secureLogin	String	User name for authentication in the Pragmatic Play service. The value will be provided by Pragmatic Play.	Required
roundId	Long	Id of the round, which is finished by a player.	Required
hash	String	Hash code of the request	Required

Response parameters

Name	Description
smResult	The details of slot machine result.
error	The error code and description for troubleshooting. This field will appear only in case of error. Successful responses should not contain it.

Example of HTTP request:

```
POST /RegulationService/ReportingAPI/Portugal/smResult/ HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&roundId=123654987&hash=438fea0a68a0e9ab49df756cc20eda8f
```

Example of the successful response:

```
{
  "smResult":
"0:5;8;1;3;1#10;10;11;11;11#4;2;10;9;4#R#10#H10#122#MV#0.01#MT#1#R#10#H10#112#MV#0.01#MT#1#MG#0.08#
",
  "betMultiplier": 20,
  "error": {
    "code": 0,
    "description": "OK"
  }
}
```

Example of error response:

```
{
  "smResult": null,
  "betMultiplier": null,
  "error": {
    "code": 2,
    "description": "Game round is not found."
  }
}
```

Error codes

Below table contains the error codes that the Pragmatic Play system will return in the response to API calls.

Code	Description
1	Authentication failed. Incorrect secure login and secure password combination (will be returned also if the hash code validation is failed)
2	Game round is not found.
3	Validation failed. One or several input parameters is not set or set incorrectly.
10	Incomplete game. Game round is not finished by the player.
100	Internal server error.

Examples

This section provides the examples of smResult field generated for various combinations. The content of the field generated by the service was formatted for easier understanding.

Paying symbol

The symbol that pays itself and is not required to be on a payline.

```
0: 1;10;8;10;9# //Game screen
12;11;5;3;12#
8;4;1;6;1#
R#1#P1#0;12;14#MV8.75#MT2# //Symbol 1 paying 8.75 and its positions
```

Winning in bonus game

Bonus game (BG) was triggered by a scatter, which is not required to be on a payline.

```
0: 3;13;13;13;3# //Game screen
3;13;3;13;3#
3;13;13;13;13#
3;3;13;13;13#
R#3#P3#0;5;10;15#MV5.5#MT2# //Symbol 3 triggered the BG, its positions and BG win amount
```

Hidden paying symbol

The symbol is paying, and it is hidden and not shown to the player explicitly because it was replaced on the game screen by an expanding symbol.

```
0: 12;12;2;9;2# //Game screen
9;9;2;10;9#
2;2;2;2;11#
R#9#H9#111#MV0.25#MT2# //Winning lines begin
R#12#H12#000#MV0.15#MT2#
R#2#H2#2222#MV2.5#MT2#
R#9#H9#210#MV0.25#MT2#
R#9#H9#12221#MV2.5#MT2#
R#12#H12#0012#MV0.5#MT2#
R#9#H9#22100#MV2.5#MT2#R
#9#H9#12101#MV2.5#MT2#
R#9#H9#211#MV0.25#MT2#
R#9#H9#212#MV0.25#MT2#
R#9#H9#110#MV0.25#MT2#
R#9#H9#112#MV0.25#MT2#
R#12#H12#002#MV0.15#MT2#
R#2#H2#2202#MV2.5#MT2#
R#12#H12#02220#MV1.25#MT2#
R#12#H12#200#MV0.15#MT2#
R#9#H9#12021#MV2.5#MT2#
R#12#H12#02020#MV1.25#MT2#
R#12#H12#202#MV0.15#MT2# //Winning lines end
R#1#P1#2;4;10#MV1.25#MT2# //Symbol 1 pays 1.25 but its positions do not match game screen
```

XVI. Integration with demo games

Demo casino game machines are a way for online casino players to get a taste for games before they sign up and play for real money. By playing before depositing money, players get better acquainted with the features and details of the game and therefore begin with an advantage. Demo games may be used by the operators in order to advertise and promote games and their brand.

The Demo Games Integration Process is simple and is described below.

In order to start using demo games, the Operator must implement games opening at the website by the way described below. The demo games can be placed on any advertisement page across the web. The format of the link for opening a demo game is as follows:

```
http(s)://{demo_games_domain}/gs2c/openGame.do?gameSymbol={game_symbol}&lang={language}&cur={currency_symbol}&jurisdiction={jurisdiction}&lobbyUrl={URL}&styleName={secureLogin}
```

Parameter	Status	Description	Example
demo_games_domain	Required	Domain configured for demo games	<i>demogamesfree.pragmaticplay.net</i>
gameSymbol	Required	A symbol that is provided for every game that is available as demo game.	vs20bs, vs25f, bjb, kna, vpa
lang	Optional	The game language. If it is not passed the 'en' will be used as default. <i>The list of available languages is described in Game Library.</i>	de, en, es, fr, it, ja, ru, th, tr, vi, zh
cur	Optional	Currency symbol that will be displayed in the game window. If it is not passed the 'USD' will be used as default. The list of available currencies is described in	USD, EUR, AUD, ZAR etc.
lobbyURL	Optional	Encoded URL for redirecting, if player clicks <i>Home</i> button. This URL may contain different parameters and should be started with <i>http(s)://</i> It is possible to pass JavaScript instead of URL: <i>js://{script}</i>	<i>http%3A%2F%2Fsomewebsite.com%2F/lobby/</i> <i>js://window.history.back();</i>
jurisdiction	Optional	Jurisdiction of the player. The list of available jurisdictions can be found in section 3.3 Data types	UK, IT, ES, LT etc.
styleName*	Optional	SecureLogin configured for real brand on PragmaticPlay side. It should be sent in order to launch Demo session with real game configurations, such as RTP, in markets where it is required by regulation. <i>In case Casino Operator needs real game configuration to be loaded in demo games with styleName, they should ask the Pragmatic Play's Technical Support for additional configuration.</i>	<i>some_secureLogin</i>

* **styleName** parameter in Demo games is not applicable for unregulated markets

Example of demo game opening link:

Asia region:

https://demogamesfree-asia.pragmaticplay.net/gs2c/openGame.do?lang=zh&cur=CNY&gameSymbol=vs243fortune&jurisdiction=UK&lobbyURL=https://www.pragmaticplay.com&stylename=some_secureLogin

Europe, rest of the world:

https://demogamesfree.pragmaticplay.net/gs2c/openGame.do?lang=en&cur=USD&gameSymbol=vs40beowulf&lobbyURL=https://www.pragmaticplay.com&stylename=some_secureLogin

XVII. Bingo Integration API

Bingo Game API

Using Bingo game API, OPERATOR is able to integrate both types of games: web versions for desktop and mobile devices. Provider provides Bingo builds based on HTML5 technology. HTML5 versions are available for desktop and mobile devices.

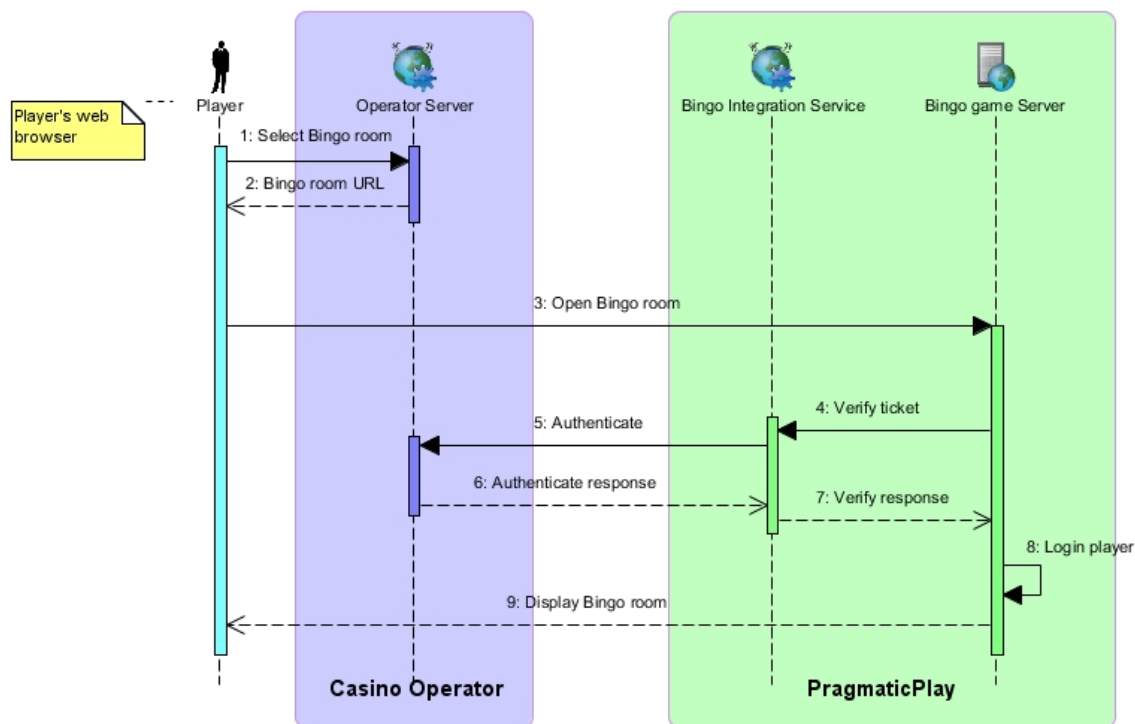
Definitions

List of definitions for the OPERATOR:

1. Bingo room – specific theme for web client.
2. Bingo game session – order of bingo games for specific period of time and bingo room.
3. Bingo game – specific settings in game for bingo game round.

17.1 Bingo room Launching

Room opening



Bingo game launch URL

In order to complete URL to launch Bingo client, operator should form URL with the following parameters and values used:

- Bingo game server domain – domain name for bingo game opening and playing;
- **key** – URL-encoded list of parameters:
 - **token** – secure one-time token is generated by OPERATOR for specific player;
 - **room** – 9999;
 - **language** – player language in ISO 639-1 standard (e.g. en, fr, it);
 - **cashierURL** – URL for opening a Cashier page from the website verified by OPERATOR;

- **lobbyURL** – URL for redirecting a player from the client to the lobby page in case if player taps/clicks on Lobby button;
- **gamesLobby** – external html page which shows user a list of mini slots and manages to launch them ([Mini Slots](#))
- **profileURL** – user profile page;
- **promotionURL** – page which includes promotion information;
- **responsibleGaming** – separate page which contains ‘responsible gaming’ info.
- **rci** – the reality check interval, in minutes. This is the interval for reality check messages to be shown to the player during game play
- **rce** – the reality check elapsed time, in minutes. This is optional parameter, which indicates already elapsed minutes on another games. If this parameter is sent by Operator, the first time RC message will appear in rci – rce minutes after game opening
- **rcHistoryUrl** – a link to the player’s game history. This is optional parameter. If Operator send the rcHistoryUrl parameter, game client will add the “History” button to the reality check message. There are two options here:
 - Operator may provide a URL to the page on their side. In this case rcHistoryUrl should contain a link to Operator’s page. For example: rcHistoryUrl=http://casinosite.com/history/
 - Game client can show inbuilt game history, provided by Pragmatic Play. In this case Operator should send the predefined value: rcHistoryUrl=ingame
- **rcCloseUrl** – a link to the page on the Operator’s website, to which the player will be redirected if they choose to close the game. This parameter is optional. The “Close” button appears on the reality check message only if Operator sends the URL
- **selfTestPageURL** – for Sweden jurisdiction. Required logos link to Operator side
- **budgetURL** – for Sweden jurisdiction. Required logos link to Operator side
- **suspensionURL** – for Sweden jurisdiction. Required logos link to Operator side
- **timer** – for Sweden jurisdiction. This value Operator can set as initial for logged time session. In case if parameter is empty, logged session will be calculated from beginning.
- **isNewWindow** – parameter to define if bingo client needs to display some elements which are related to full tab view. (ex logo in header). Available values:
 - true – new window
 - false – iframe inside OPERATOR’s website
- **stylename** – unique identifier of the OPERATOR at PROVIDER side.

Example:

<https://<Server domain>/bingo/playBingo.do?key=token%3D<token>%26room%3D<9999>%26language%3D<language>%26cashierURL%3D<cashierURL>%26lobbyURL%3D<lobbyURL>%26profileURL%3D<profileURL>%26preBuyCardURL%3D<preBuyCardURL>%26promotionURL%3D<promotionURL>%26rci%3D0%26rce%3D0%26rcHistoryUrl%3D<rcHistoryUrl>%26rcCloseUrl%3D<rcCloseUrl>%26gamesLobby%3D<gamesLobbyURL>%26responsibleGaming%3D<responsibleGamingURL>%26selfTestPageURL%3D<OperatorUrl>%26budgetURL%3D<OperatorUrl>%26suspensionURL%3D<OperatorUrl>%26timer%3D<playedPeriod>%26isNewWindow%3Dfalse&stylename=<operator stylename>>

Desktop version of bingo client should have minimal width for wrapper (iframe) 840dp and height 500dp, so that content can be fully displayed. In **mobile** version, for phones and tablets, the bingo client support min window width 320dp.

Our Backend allows to disable some elements in bingo client depending on its version:

1. For **Mobile** version:
 - Multi Tabs – no switching between rooms within mobile bingo client interface;
 - Balance – no balance with deposit button within mobile bingo client interface;
 - Menu – no menu button within mobile bingo client interface;

- Mini-games lobby – no mini games functionality within mobile bingo client interface;
- Chat – no chatting functionality;
- 2. For **Desktop** version:
 - Header – no header within desktop bingo client interface replacing balance and menu to interface below, no bingo lobby functionality;
 - Menu - no menu button within desktop bingo client interface;
 - Balance – no balance field within desktop bingo client interface;
 - Minigames - no mini games functionality within desktop bingo client interface;
 - Chat - no chat area within desktop bingo client interface;

In order to access different settings and launch bingo client inside iframe on Operator websites, the following cases with custom messages are available:

- { name: 'notifyCloseContainer' } - is sent for any closing event inside bingo client which notifies main window that container with bingo client should be closed.
- in case if Operator requires room list filtration for lobby inside bingo client or preorders, by PP back office setting will be activated automatically sending for cases when client has launched post message { type: 'getRooms' }. After this Operator's site should response with post message { type: 'rooms', roomId: [...] }, with rooms which are available for current user. Each time when site list is updated, the new post message should be sent.
- { name: 'nonAuthorized' } - this post message is sent to main window each time when non-logged user is trying to open a room from bingo client, that site could show login form above bingo client.
- { type: 'gameHistory', url: '/gameHistory/' } - is sent to main window = if Menu within bingo client interface is disabled and Operator wants to use user bingo history from PP.
- { type: 'deposit' } - is sent to main window for deposit button action inside bingo client interface, if Operator requires keep bingo client and deposit window in same tab. Need to be configured in PP back office.

Mini-games

Bingo client allows OPERATOR to introduce user to Bingo, video slots and other games by using iframe in special area inside web application.

In order to do that, the OPERATOR needs to create a special lobby page for mini-games list (should be hosted on OPERATOR side), which will allow player to choose and play any mini-game inside Bingo client.

Iframe's size for lobby page in the **desktop version** should have non-fixed height but alterable width between 720dp and 480dp. Mini-game will be launched in special container as separate iframe, by post message notification from lobby-page to main window, in moment when user clicks to launch a min-game. The aspect ratio between width and height is 16:9 for mini-game container in each resolution.

Iframe's size for lobby page in the **mobile version** has alterable width between 320dp and higher, depends on device display size, an aspect ratio of iframe between width and height is 16:9.

OPERATOR needs to implement special post messages logic for Bingo web client page.

Implementation of post messages by the dispatched event explanation:

'otherWindow' can listen to dispatched messages by executing the following JavaScript:

```
window.addEventListener("message", receiveMessage, false);

function receiveMessage(event)
{
  if (event.origin !== "http://example.org:8080")
    return;
  // ...
}
```

NOTE: External link from the Operator side should always begin with HTTPS. CORS headers should always be present.

Example of implementation on OPERATOR side:

```
<div class="mini-games__item" >
  <div
    class="mini-games__item-img"
    style="background-image: url(/resources/mini-slots/madame-destiny.png)"
    data-link="https://demogamesfree.pragmaticplay.net/gs2c/openGame.do?gameSymbol=vs10madame&minimode=1"
  ></div>
</div>
```

Bingo client expects receive next:

`window.postMessage(JSON.stringify({ type: 'OPEN_SLOT', link: <link of launcher>}), '*');` – when game is launching on OPERATOR lobby page should be sent such post message.

For game Providers of mini-games to sync updating of balance between bingo and game, we recommend to support next post message: `postMessage({ name: 'updateBalance' }, '*')`. This post message will be sent by bingo client to child window with mini-game container to update balance information.

For more information use the following documentation:

<https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage>

17.2 Bingo integration API for OPERATOR side

This is a simple API for Pragmatic Play gaming platform to connect to player's wallet. API is an HTTP listener, which listens application/x-www-form-urlencoded POST requests.

To connect player's wallet at operator side, methods and specification which are described in section [III Seamless Wallet API](#) of current document should be used.

As an option, In order to set player's nickname which will be used in bingo chat, provider can pass optional parameter "nickname" in response of method [3.4 Authenticate](#).

Nickname

Request path: /nickname.html

**request path is the same as for Seamless Wallet API*

This method is used by PragmaticPlay side for informing OPERATORS about what nickname is used by player in bingo system. For example in situation when Operator doesn't support nickname parameter in Authenticate method, user needs to set up nickname inside bingo game client, and this nickname will be forwarded back to OPERATOR in this method.

Request parameters

Name	Type	Description	Status
userId	String(100)	Identifier of the user within the Casino Operator system.	Required
nickName	String(100)	Current nickname of player in Pragmatic Play system.	Required
providerId	String (32)	Game provider identifier.	Required
hash	String (32)	Hash code of the request	Required

Response parameters

Name	Type	Description	Status
error	Number	Response status.	Required
description	String(250)	Response status short description.	Required

Example of HTTP request:

```
POST /nickname.html HTTP/1.1
Host: someoperatordomain.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

nickName=PlayerNickname&userId=16b31c8848b21&providerId=PragmaticPlay&hash=d8450312aee2fe68c76f80be75d26849
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
}
```

Free Round Bonus Prizes

This feature allows operator to set prize type as Free Round Bonuses (FRB), which further can be used by player in PragmaticPlay slot games.

In order to support Spinner room with FRB prizes in BINGO product, In **3.8 Result**, **3.11 EndRound**, **3.9 BonusWin** methods will be added new optional array **specPrizes[#]** with next parameters inside:

- specPrizeAmount – number of FRB which was won by player in bingo game
- specPrizeCode – unique FRB bonus code within PP system per bingo game and player.
- specPrizeType – “FRB” which will be special mark that this prize is Free rounds bonus for casino game

This list is optional and is not sent to the Casino Operator by default. In case Casino Operator needs this parameter to be sent with request they should ask the Pragmatic Play’s Technical Support for additional configuration. Such result should be sent only if player won FRB with other monetary prize. Otherwise if players has only FRB prize in current game, this list should be taken from endRound and result is not sent. If Operator’s players participate in bingo games with FRB prizes but this list is not set up, result won’t be sent at all and round will be closed without winners on Operator side.

17.3 Bingo integration API for PROVIDER side

Hash calculation

In Provider side API security hash is passed as special request header x-hash

Consider the RoomList request as an example of hash calculation.

NOTE: parameter names should be taken for calculation by order from json request

Phase 1: Get request body

Example:

```
{
  "language": "en",
  "playerID": "1677d81655c46",
  "secureLogin": "secure login"
}
```

Phase 2: Add 'secure password' to request body string

Example:

```
{
  "language": "en",
  "playerID": "1677d81655c46",
  "secureLogin": "secure_login"
}secure_password
```

Phase 3 Encode MD5

Example:

[62e80083e2ffb016890d33aae54a240d](#)

Phase 4: Add x-hash header to the request

Example:

```
POST /BingoIntegration/BingoGameAPI/RoomList HTTP/1.1
Host: api-domain
-H 'x-hash: 62e80083e2ffb016890d33aae54a240d'
-H 'Content-Type: application/json'
-d '{
  "language": "en",
  "playerID": "1677d81655c46",
  "secureLogin": "secure_login"
}'
```

RoomList

This method returns a list of active bingo rooms for the current time period.

Request parameters

Property	Type	Description	Status
playerID	string	Unique player identifier	Optional
language	String	Player's language by ISO 639-1	Optional
secureLogin	String	Login to access the API	Required

Example of request:

```
POST /BingoIntegration/BingoGameAPI/RoomList HTTP/1.1
Host: api-bingo.prerelease-env.biz
x-hash: 3a41716c7db9a10313ff01b17ebad3f4
Content-Type: application/json
{"secureLogin":"secureLogin","playerID":"91449116","language":"en"}
```

Response parameters

Property	Type	Description
error	integer	Response status.
description	string	Response status - short description.
activeRoomList	array of 'ActiveRoom' type	List of active bingo rooms.
upcomingRoomList	array of 'UpcomingRoom' type	List of upcoming bingo rooms.

Structure 'ActiveRoom' & Structure 'UpcomingRoom'

Property	Type	Description
roomID	long	Bingo room identifier.
bingoType	integer	Bingo game type
roomName	string	Bingo room name.
cardCost	array of type 'CurrencyEntry'	Card price.
gamePrize	array of type 'CurrencyEntry'	Game prize amount.
nextGameStart	number	Starting time of the next game. (in UNIX timestamp in ms)
jackpotAmount	array of type 'CurrencyEntry'	Jackpot (if configured).
playersBoughtCards	integer	Number of players who bought cards for the current game.
playersInRoom	integer	Number of players in the current bingo room.
jurisdiction	string	May contain one or multiple jurisdictions comma separated
inSession	boolean	Possible values - true, false. If true - the tickets pre-order is available only for full session (for pack of all Bingo rounds inside the session)
private	boolean	Possible values - true, false. If true - the room has specific entry conditions. If false – regular public room.
x2goPrize	array of type 'XtoGO prizes'	XtoGO prizes
rollOnPrizes	Array of the type 'rollOnPrizes'	Array of Roll on prizes

Structure 'XtoGO prizes'

Property	Type	Description
x2go	integer	x2go level. Possible values - 1-10
prize	array of type 'CurrencyEntry'	x2go prize amount.

Structure 'rollOnPrizes'

Property	Type	Description
rollOn	Number	Level of Roll on prize, Possible values 1-20
prize	array of type 'CurrencyEntry'	Pot for each XtoGO prize

Structure 'CurrencyEntry'

Property	Type	Description
symbol	string	Currency symbol by ISO 4217 (USD, AUD, EUR etc)
value	double	Amount in relevant currency.
prizeType	string	FRB – for FRB prizes, this field present only when FRB prizes
betPerLine	Number	Value of money for one line in casino games. This field present only when FRB prizes

NOTE: In one game can be either Xtogos or Roll-ons as additional prizes, or nothing

Example of response

```
{
  "error": 0,
  "description": "OK",
  "activeRoomList": [
    {
      "roomID": 1,
      "bingoType": 75,
      "roomName": "Room 1",
      "cardCost": [
        {
          "symbol": "USD",
          "value": 0.1
        }
      ],
      "gamePrize": [
        {
          "symbol": "USD",
          "value": 100.0
        }
      ],
      "nextGameStart": 1543999992208,
      "jackpotAmount": [
        {
          "symbol": "USD",
          "value": 1000.0
        }
      ],
      "playersBoughtCards": 10,
      "playersInRoom": 10,
      "jurisdiction": "99",
      "inSession": false,
      "x2goPrizes": [
        {
          "x2go": 1,
          "prize": [
            {
              "symbol": "USD",
              "value": 1
            },
            {
              "symbol": "EUR",
              "value": 0.8
            }
          ]
        }
      ],
      "private": false
    },
    {
      "roomID": 2,
      "bingoType": 90,
      "roomName": "Room 2",
      "cardCost": [
        {
          "symbol": "USD",
          "value": 0.1
        }
      ],
      "gamePrize": [
        {
          "symbol": "USD",
          "value": 100.0
        }
      ],
      "nextGameStart": 1543999992208,
      "jackpotAmount": [
        {
          "symbol": "USD",
          "value": 1000.0
        }
      ],
      "playersBoughtCards": 10,
      "playersInRoom": 10,
      "jurisdiction": "99",
      "inSession": false,
      "x2goPrizes": [
        {
          "x2go": 1,
          "prize": [
            {
              "symbol": "USD",
              "value": 1
            },
            {
              "symbol": "EUR",
              "value": 0.8
            }
          ]
        }
      ],
      "private": false
    }
  ]
}
```

AvailableGames

The following method returns a list of rooms with a games schedule (when which game will be played). Also, it returns information about the player's pre-ordered cards. Interval between **dateFrom** and **dateTo** should be equal, or 24 hours, or 1 hour, otherwise it will be rounded to these values.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API	Required
dateFrom	long	Start date of the request. (UNIX time in ms)	Required
dateTo	long	End date of the request. (UNIX time in ms)	Required
playerID	String	Unique player identifier	Required
language	String	Player's language by ISO 639-1	Optional
roomID	integer	Bingo room identifier.	Optional

Example of request

```
POST /BingoIntegration/BingoGameAPI/AvailableGames HTTP/1.1
Host: api-bingo.prerelease-env.biz
x-hash: 496e9245d50e2b4c27de17216dc831cc

{"secureLogin": "securelogin", "language": "en", "playerID": "1623e6952764", "dateFrom": 1537574400000, "dateTo": 1537578000000}
```

Response parameters

Property	Type	Description
error	integer	Response status.
description	String	Response status - short description
roomList	array of type 'Room'	All rooms scheduled for the current brand.

Structure 'Room'

Property	Type	Description
roomID	integer	Bingo room identifier.
jurisdiction	string	jurisdiction ID of the current room
roomName	String	Bingo room name.
gameList	array of type 'Game'	List of games in the room during specified time period

Structure 'Game'

Property	Type	Description
gameID	string	Bingo game identifier. (Bingo game ID)
gameName	string	Name of Bingo game.
timeGameStart	long	Start date of the request. (UNIX time in ms)

serialGameIdx	integer	Serial number of the game at the requested hour. This parameter is required for the accurate determination of the necessary game in an hour, so as the game can be repeated many times.
maxCardsNumber	integer	Max number of cards available for purchase.
cardCost	array of type 'CurrencyEntry'	Card price
purchaseMode	String	C – Card / Ticket F – Fixed price S – Strips Fixed price T – Strips Ticket price
players	integer	Number of players who have already preordered cards for the game.
boughtCardsNumber	integer	Number of cards preordered by player.
givenCardsNumber	integer	Number of free cards that were received as bonus with preordered cards.
totalAmount	array of type 'CurrencyEntry'	Total amount spent by the player on cards preordering.
gamePrize	array of type 'CurrencyEntry'	Game prize amount.
jackpotAmount	array of type 'CurrencyEntry'	Jackpot POT amount.
bingoType	integer	Bingo game type

Structure 'CurrencyEntry'

Property	Type	Description
symbol	string	Currency symbol by ISO 4217 (USD, AUD, EUR etc)
value	double	Amount in relevant currency.
prizeType	string	FRB – for FRB prizes, this field present only when FRB prizes
betPerLine	Number	Value of money for one line in casino games. This field present only when FRB prizes

Example of response

```
{
  "error": 0,
  "description": "OK",
  "roomList": [
    {
      "roomID": 2163,
      "roomName": "Vitalis Room",
      "gameList": [
        {
          "gameID": "5723;1;1990",
          "gameName": "game 1 Name",
          "serialGameIdx": 1,
          "maxCardsNumber": 100,
          "purchaseMode": "C",
          "players": 11,
          "boughtCardsNumber": 221,
          "givenCardsNumber": 0,
          "cardCost": [
            {
              "symbol": "EUR",
              "value": 1.63
            },
            {
              "symbol": "USD",
              "value": 2.0
            }
          ],
          "totalAmount": [
            {
              "symbol": "EUR",
              "value": 1.63
            },
            {
              "symbol": "USD",
              "value": 2.0
            }
          ],
          "timeGameStart": 1552644600000,
          "gamePrize": [
            {
              "symbol": "EUR",
              "value": 16.3
            },
            {
              "symbol": "USD",
              "value": 20.0
            }
          ],
          "jackpotAmount": [
            {
              "symbol": "EUR",
              "value": 1630.0
            },
            {
              "symbol": "USD",
              "value": 2000.0
            }
          ],
          "bingoType": 75
        }
      ],
      "jurisdiction": "UK"
    }
  ]
}
```


PreorderCards

The following method provides the possibility to make cards pre-orders.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API	Required
playerID	String	Unique player identifier	Required
nickname	string	Player's nickname.	Optional
currency	String	Player's currency.	Required
roomID	integer	Bingo room identifier.	Required
preorderCardList	array of type 'PreorderCard'	List of pre-ordered cards for selected games.	Required

Structure 'PreorderCard'

Property	Type	Description	Status
gameID	string	Bingo game identifier. (Bingo game ID)	Required
serialGameIdx	integer	Serial index of the game at the requested hour. This parameter is required for the accurate determination of game queue position in an hour, as the game can be played many times in the same hour.	Required
cardNumber	integer	Number of pre-ordered cards.	Required
timeGameStart	long	Start date of the request. (UNIX time in ms)	Required
authToken	string	Authenticate token to /authenticate.html request. Used for room opening and making a BET	Optional

Example of request

```
POST /BingoIntegration/BingoGameAPI/PreorderCards HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab

{"playerID":"1623e6952764","nickname":"Vitalii","currency":"USD","roomID":1,"preorderCardList":[{"gameID":1050,"serialGameIdx":2,"cardNumber":50,"timeGameStart":1556204400000}],"secureLogin":"secureLogin"}
```

Response parameters

Property	Type	Description
error	string	Response status.
description	string	Response status - short description
preorderCardList	array of type 'PreorderCard'	List of pre-ordered cards with processing results.

Structure 'PreorderCard'

Property	Type	Description
error	integer	Response status.
description	string	Response status - short description
gameID	integer	Bingo game identifier. (Bingo game ID)
serialGameIdx	integer	Serial index of the game at the requested hour. This parameter is required for the accurate determination of game queue position in an hour, as the game can be played many times in the same hour.
cardNumber	integer	Number of pre-ordered cards.
givenCardsNumber	integer	Number of free cards that were received as bonus with preordered cards.
timeGameStart	Number	Start date of the request. (UNIX time in ms)

Example of response

```
{
  "error": 0,
  "description": "OK",
  "preorderCardList": [
    {
      "error": 0,
      "description": "OK",
      "gameID": 6940,
      "serialGameIdx": 13,
      "cardNumber": 10,
      "givenCardsNumber": 0,
      "timeGameStart": 1556630993000
    }
  ]
}
```

AvailableSessions

This method returns available promo sessions for next 30 days to the requested date for the room which has the flag **"InSession=true"** in the RoomList method.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API	Required
playerID	String	Unique player identifier.	Required
fromDate	long	Start date of the report. (UNIX time in ms)	Required
language	String	Player's language by ISO 639-1	Optional
roomID	long	Bingo room identifier.	Required

Example of request

```
POST /BingoIntegration/BingoGameAPI/AvailableSessions HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab
{"roomID":2500,"fromDate":61499157200000,"playerID":"1623e6952764","language":"en","secureLogin":"vitalis"}
```

Response parameters

Property	Type	Description
error	integer	Response status.
description	string	Response status - short description
availableSessions	array of type 'availableSessions'	List of sessions available for preorder in next 30 days

Structure 'availableSessions'

Property	Type	Description
status	string	Status of the current session (played, inPlay, upcoming)
sessionID	Long	Unique session Identifier
name	string	Name of the session in player's language
players	integer	Number of players who bought tickets during the session
price	array of type 'CurrencyEntry'	Card price inside the session in player's currency
discountAmount	array of type 'CurrencyEntry'	Discount applied to cards price inside the session
nickname	string	Winner's nickname (if session has been played)
prize	array of type 'CurrencyEntry'	Winner's prize (if session has been played)
sessionStart	long	Session start date and time (UNIX timestamp)
numberOfGames	integer	Number of games inside the session
numberOfTickets	integer	Number of tickets available inside the session
purchased	Boolean	True or false. True if the user has purchased cards for the session

Example of response

```
{
  "error": 0,
  "description": "OK",
  "availableSessions": [
    {
      "status": "played",
      "sessionID": 1,
      "name": "MORNING",
      "players": 500,
      "price": [
        {
          "symbol": "EUR",
          "value": 1.5
        },
        {
          "symbol": "USD",
          "value": 1.0
        }
      ],
      "discountAmount": [
        {
          "symbol": "EUR",
          "value": 1.5
        },
        {
          "symbol": "USD",
          "value": 1.0
        }
      ],
      "nickname": "PlayerNickname",
      "prize": [
        {
          "symbol": "EUR",
          "value": 1.5
        },
        {
          "symbol": "USD",
          "value": 1.0
        }
      ],
      "sessionStart": 61499167200000,
      "numberOfGames": 36,
      "numberOfTickets": 3600,
      "purchased": null
    },
    {
      "status": "inPlay",
      "sessionID": 2,
      "name": "AFTERNOON",
      "players": 500,
      "price": [
        {
          "symbol": "EUR",
          "value": 1.5
        },
        {
          "symbol": "USD",
          "value": 1.0
        }
      ],
      "discountAmount": [
        {
          "symbol": "EUR",
          "value": 1.5
        },
        {
          "symbol": "USD",
          "value": 1.0
        }
      ],
      "nickname": null,
      "prize": [
        {
          "symbol": "EUR",
          "value": 1.5
        },
        {
          "symbol": "USD",
          "value": 1.0
        }
      ],
      "sessionStart": 61499177200000,
      "numberOfGames": 36,
      "numberOfTickets": 3600,
      "purchased": null
    },
    {
      "status": "upcoming",
      "sessionID": 3,
      "name": "EVENING",
      "players": 50,
      "price": [
        {
          "symbol": "EUR",
          "value": 1.5
        },
        {
          "symbol": "USD",
          "value": 1.0
        }
      ],
      "discountAmount": [
        {
          "symbol": "EUR",
          "value": 1.5
        },
        {
          "symbol": "USD",
          "value": 1.0
        }
      ],
      "nickname": null,
      "prize": [
        {
          "symbol": "EUR",
          "value": 1.5
        },
        {
          "symbol": "USD",
          "value": 1.0
        }
      ],
      "sessionStart": 61499187200000,
      "numberOfGames": 36,
      "numberOfTickets": 3600,
      "purchased": false
    }
  ]
}
```

SessionGameInfo

This method returns a list of games inside a specific session.

Request parameters

Property	Type	Description	Status
secureLogin	string	Login to access the API	Required
playerID	string	Unique player identifier.	Required
promoSessionStart	Number	Start date of the session. (UNIX time in ms)	Required
language	string	Player's language by ISO 639-1	Required
sessionID	Number	Unique session Identifier	Required

Example of request

```
POST /BingoIntegration/BingoGameAPI/SessionGameInfo HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab

{"secureLogin":"7casino","playerID":"1638c84a9e857","promoSessionStart":1587988800000,"sessionID":482,"language":"en"}
```

Response parameters

Property	Type	Description
status	string	Current session status: <ul style="list-style-type: none">played (rounds in session were finished)inPlay (there are still unfinished rounds)upcoming (sessions is going to start)
sessionStart	long	UNIX timestamp when session starts
sessionFinish	long	UNIX timestamp when session ends
players	integer	Number of players who bought tickets in this session
name	string	Session name
price	array of type 'CurrencyEntry'	Common price of cards for all rounds in session
purchased	Boolean	True or false. True if user has purchased cards for this session
sessionGames	Array	List of games (rounds) inside the current session. Array SessionGame
error	integer	Response status.
description	string	Response status - short description

Structure 'SessionGame'

Property	Type	Description
bingoType	integer	Bingo game type
gameStart	long	Timestamp of the game starts.
tickets	integer	Tickets of the current game.
prize	list	Prize by game. Array of type [CurrencyEntry].
jackpotType	string	Type of jackpot (optional).
jackpotAmount	list	Jackpot amount (optional). Array of type [CurrencyEntry].

Example of response

```
{ "error": 0, "description": "OK", "status": "upcoming", "sessionStart": 1588788000000, "sessionFinish": 1588791599000, "players": 0, "name": "promo", "price": [ { "symbol": "EUR", "value": 270.0 }, { "symbol": "USD", "value": 540.0 } ], "purchased": false, "sessionGames": [ { "bingoType": 75, "gameStart": 1588788000000, "tickets": 100, "prize": [ { "symbol": "EUR", "value": 0.0 }, { "symbol": "USD", "value": 0.0 } ], "jackpotType": "CFJ", "jackpotAmount": [ { "symbol": "EUR", "value": 200.0 }, { "symbol": "USD", "value": 400.0 } ] } ] }
```

BuySession

This method returns a list of cards preordered by the player.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API.	Required
playerID	String	Unique player identifier.	Required
promoSessionStart	long	(Unix time) Start of the session. Value received in response to the AvailableSessions request.	Required
sessionID	integer	Unique session Identifier	Required
authToken	String	Authenticate token to /authenticate.html request. Used for room opening and making a BET	Optional

Example of request

```
POST /BingoIntegration/BingoGameAPI/ BuySession HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab

https://{bingo API domain}/BingoIntegration/BingoGameAPI/ BuySession
```

Response parameters

Property	Type	Description	Status
error	integer	Response status.	Required
description	string	Response status - short description.	Required

Example of response

```
{"error":0,"description":"OK"}
```

PreorderedReport

This method returns a list of cards preordered by the player.

Request parameters

Property	Type	Description	Status
secureLogin	string	Login to access the API	Required
playerID	string	Unique player identifier.	Required
dateFrom	long	Start date of the report. (UNIX time in ms)	Required
dateTo	long	End date of the report. (UNIX time in ms)	Required
language	string	Player's language by ISO 639-1	Optional
roomID	integer	Bingo room identifier.	Optional

Example of request

```
POST /BingoIntegration/BingoGameAPI/PreorderedReport HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab
```

```
{
  "secureLogin": "securelogin",
  "playerID": "16243a2ec4218f2",
  "dateFrom": 1544433758165,
  "dateTo": 1544606558165,
  "language": "no",
  "roomID": 1442
}
```

Response parameters

Property	Type	Description
error	integer	Response status.
description	string	Response status - short description.
preorderedList	array of type 'PreorderedReport'	List of games which have preordered cards for specific timeframe

Structure 'PreorderedReport'

Property	Type	Description
transactionID	long	Reference of bet request in wallet API.
sessionStartDate	long	Date and time of the draw phase. (UNIX time in ms)
gameID	integer	Bingo game identifier. (Bingo game ID)
gameName	string	Name of Bingo game.
bingoType	integer	Bingo game type
roomID	integer	Bingo room identifier.
roomName	string	Bingo room name.
cardNumbers	integer	Number of cards.

givenCardsNumber	integer	Number of cards given to players as bonus (relevant to Buy X get Y games).
cardCost	double	Card price.
purchaseMode	string	C – Card / Ticket F – Fixed price S – Strips Fixed price T – Strips Ticket price
roundID	long	ID of a round with cards preordered by the player
bingoRoundID	long	Unique identifier of Bingo round from Provider side where player will play preordered cards.
currency	string	Player's currency. (ISO 4217)
prize	double	Prize amount for winning a round (in player's currency)
jackpot	double	Jackpot amount (in player's currency)
prizeType	string	FRB – for FRB prizes, this value present only when special prizes
betPerLine	Number	Value of money for one line in casino games for player currency. This value present only when FRB prizes

Example of response

```
{
  "error": 0,
  "description": "OK",
  "preorderedList": [
    {
      "sessionStartDate": 1544608800000,
      "gameID": 6000,
      "gameName": "bingo_90_jp",
      "bingoType": 90,
      "roomID": 1442,
      "roomName": "for test with 90",
      "cardNumbers": 20,
      "cardCost": 1,
      "purchaseMode": "C",
      "roundID": 20007000207,
      "currency": "USD",
      "transactionID": 20001661407,
      "bingoRoundID": 71761061,
      "givenCardsNumber": 0,
      "prize": 100.11,
      "jackpot": 0
    },
    {
      "sessionStartDate": 1544608800000,
      "gameID": 6000,
      "gameName": "bingo_90_jp",
      "bingoType": 90,
      "roomID": 1442,
      "roomName": "for test with 90",
      "cardNumbers": 20,
      "cardCost": 1,
      "purchaseMode": "C",
      "roundID": 20007000207,
      "currency": "USD",
      "transactionID": 20001661607,
      "bingoRoundID": 71761061,
      "givenCardsNumber": 0,
      "prize": 100.11,
      "jackpot": 0
    }
  ]
}
```

JackpotWinners

This method returns jackpot winners with all required information for specific date.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API	Required
dateFrom	Number	Start date of the request. UNIX time in ms	Required
dateTo	Number	End date of the request. UNIX time in ms	Required
language	String	Player's language	Optional

Example request

```
POST /BingoIntegration/BingoGameAPI/JackpotWinners HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab
{"dateFrom":1544054400000,"dateTo":1544140800000,"language":"en","secureLogin":"vitalis"}
```

Response parameters

Property	Type	Description	Status
error	Number	Response status.	Required
description	String	Response status short description.	Required
jackpotWinners	array of type 'JackpotWinner'	The list of objects for transferring information about jackpot winners.	Required

Structure 'JackpotWinner'

Property	Type	Description	Status
playerID	string	Unique player identifier.	Required
nickname	string	Player's nickname.	Optional
amount	double	Jackpot won amount.	Required
dateTime	number	Date and time when jackpot is won. UNIX time in ms	Required
jackpotName	string	Name of the jackpot.	Required
roomID	number	ID of the room.	Required
bingoType	integer	Bingo game type (75,90,80,50,30)	Required
gameName	string	Name of the game when jackpot is won.	Required
roundID	number	ID of the round.	Required
currency	string	Player's currency ISO 4217	Required

Example response

```
{
  "error": 0,
  "description": "OK",
  "jackpotWinners": [
    {
      "playerID": "169d886556e2",
      "nickname": "some_nickname",
      "amount": 11.7,
      "dateTime": 1556286943000,
      "jackpotName": "progressive",
```

```
    "roomID": 1582,  
    "bingoType": 75,  
    "gameName": "progressive jp game",  
    "bingoRoundID": 75741724,  
    "currency": "USD"  
  },  
  {  
    "playerID": "169d886556e2",  
    "nickname": "some_nickname",  
    "amount": 117,  
    "dateTime": 1556284044000,  
    "jackpotName": "static",  
    "roomID": 1582,  
    "bingoType": 75,  
    "gameName": "static jp game",  
    "bingoRoundID": 75741428,  
    "currency": "USD"  
  }  
]  
}
```

ChangeNickname

This method is responsible for changing of player's nickname.

Request parameters

Property	Type	Description	Status
secureLogin	string	Login to access the API	Required
playerID	string	Unique player identifier.	Required
nickname	string	New nickname of the player.	Required

Example request

```
POST /BingoIntegration/BingoGameAPI/ChangeNickname HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 06e0c41632d208c5b5b8fc3487279b99
```

```
{"playerID": "playerID", "nickname": "NewNickname", "secureLogin": "secureLogin"}
```

Response parameters

Property	Type	Description	Status
Error	number	Response status.	Required
description	string	Response status short description.	Required

Example response

```
{
  "error": 0,
  "description": "OK"
}
```

Bingo Player History

This method returns list of rounds that were played by player in specified period of time. It is available in BingoGameAPI.

Request parameters

Property	Type	Description	Status
secureLogin	string	Login to access the API	Required
playerID	string	Unique player identifier	Required
startDate	number	Starting date of request period. UNIX time in ms	Required
endDate	number	End date for request period. UNIX time in ms	Required
Language	string	Player's language	Required
roundid	Number	Round ID for player transaction from common API.	Optional
bingoRoundID	Number	Unique identifier of played bingo round from Provider side.	Optional
fromIndex	number	Starting index for result order in response	Required
pageOffset	integer	Number of result on page (limit 50 items)	Required

NOTE: In this request can be sent one of 2 parameters: roundID or bingoRoundID or without in case search for timeframe

Example request:

```
POST /BingoIntegration/BingoGameAPI/BingoPlayerHistory HTTP/1.1
```

```
Host: api-bingo.prerelease-env.biz
```

```
x-hash: 146ad89345eca759410c31ab3d72458b
```

```
Content-Type: application/json
```

```
{"secureLogin":"securelogin","playerID":"16243a2ec4218f2","startDate":1543660263791,"endDate":1544610663791,"language":"es","roundID":20006700607,"fromIndex":0,"pageOffset":50}
```

Response values

Property	Type	Description	Status
error	string	Number of errors.	Required
description	string	Description of error	Required
bingoRounds	array of type 'Rounds'	List of rounds.	Required
count	long	Number of founded results	Optional

Structure 'Rounds'

Property	Type	Description	Status
bingoRoundID	long	Unique identifier of played bingo round from Provider side.	Required
roomName	string	Name of room where the round was played	Required
roundDate	number	Date and time of round start. UNIX time in ms	Required
price	double	Sum of money which was spent by player	Required
cardsNumber	integer	Sum of cards which were bought for round	Required
givenCardsNumber	Integer	The number of cards given to player in addition (relevant for Buy X get Y games).	Required
freeCardsNumber	integer	The number of cards, which are played by player as free cards from FRB.	Required
prize	double	Sum of money which player won	Required

Example response:

```
{
  "error": 0,
  "description": "OK",
  "bingoRounds": [
    {
      "bingoRoundID": "71635203",
      "roomName": "vcb wwg skyl 75 vcb es",
      "roundDate": 1544087429000,
      "price": 10,
      "cardsNumber": 10,
      "givenCardsNumber": 10,
      "freeCardsNumber": 10,
      "prize": 506.01
    }
  ],
  "count": 1
}
```

Bingo Round History

This method returns information about bought / given cards for specified bingo game as well as information about played bingo games. It can be also used for receiving information about number of cards in a game.

Request parameters

Property	Type	Description	Status
secureLogin	string	Login to access the API	Required
roundID	number	Round ID for player transaction from wallet API.	Required
bingoRoundID	number	Unique identifier of played bingo round from Provider side.	Required
language	string	Player's language.	Optional

NOTE: In this request should be sent at least one of 2 parameters: roundID or bingoRoundID

Example request:

```
POST /BingoIntegration/BingoGameAPI/BingoRoundHistory HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 448f27e631b836c4549b49d492fb3423
```

```
{"roundID": 1000, "language": "en", "secureLogin": "securelogin"}
```

Response parameters

Property	Type	Description	Status
gameName	string	Name of the game.	Optional
gameID	Number	ID of the game. (Bingo game ID)	Required
roomID	Number	ID of the room.	Required
bingoType	integer	Bingo game type (75 or 90)	Required
bingoRoundID	number	Unique identifier of played bingo round from Provider side.	Required
bingoCalls	string	Called numbers divided by whitespaces up to winning call.	Optional
bingoWinners	array of BingoWinners	The list of objects with information about pattern and winners per game part. For usual game this list contain single object.	Required
boughtCards	array of type 'BoughtCards'	The list of objects with information about players and cards number, bought for specific round.	Required
error	number	Response status.	Required
description	string	Response status short description.	

Structure 'BingoWinner'

Property	Type	Description	Status
partNumber	integer	Number of part for which winners are returned.	Required
pattern	string	Name of the pattern for current part.	Required
winningCallNumber	integer	Number of call when prize was won.	Required
winnerDetails	array of 'BingoWinnerDetails'		Required

Structure 'BingoWinnerDetails'

Property	Type	Description	Status
playerID	string	Unique player identifier on operator's side	Required
currency	string	Currency of player	Required
prizeAmount	double	Prize amount won	Required
jpAmount	double	Jackpot amount won	Required
cardID	number	ID of the winning card	Required
nickname	String	Player nickname.	Required

Structure 'BoughtCards'

Property	Type	Description	Status
playerID	string	Unique player identifier.	Required
boughtCardsNumber	integer	The number of cards bought for specific round by player. For free games it is number of cards, given automatically to player during purchase period. For fixed-price game it will be number of cards in a pack.	Required
givenCardsNumber	integer	The number of cards given to player in addition (relevant for Buy X get Y games).	Required
freeCardsNumber	Integer	The number of cards, which are played by player as free cards from FRB.	Required
cardCost	double	Cards cost for specific bingo round. For fixed-price game it will be cost of one card from pack.	Required
currency	string	Currency of the player (ISO 4217). Examples: EUR' or USD'.	Optional

Example response:

```
{
  "error": 0,
  "description": "OK",
  "gameName": "vcv wvg skyl game grid it",
  "bingoType": 75,
  "bingoRoundID": 71635203,
  "gameID": 5800,
  "roomID": 1202,
  "bingoCalls": "22, 68, 55, 9, 73, 60, 71, 25, 65, 32, 39, 12, 38, 13, 28, 21, 56, 61, 41, 11",
  "bingoWinners": [
    {
      "partNumber": 1,
      "pattern": "Crazy Any 3 Corners",
      "winningCallNumber": 20,
      "winnerDetails": [
        {
          "playerID": "16243a2ec4218f2",
          "nickname": "sk_en_65",
          "currency": "USD",
          "prizeAmount": 1.67,
          "jpAmount": 167,
          "cardID": 79000
        },
        {
          "playerID": "16243a2ec4218f2",
          "nickname": "sk_en_65",
          "currency": "USD",
          "prizeAmount": 1.67,
          "jpAmount": 167,
          "cardID": 79006
        },
        {
          "playerID": "16243a2ec4218f2",
          "nickname": "sk_en_65",
          "currency": "USD",
          "prizeAmount": 1.67,
          "jpAmount": 167,
          "cardID": 79005
        }
      ]
    }
  ],
  "boughtCards": [
    {
      "playerID": "16243a2ec4218f2",
      "boughtCardsNumber": 10,
      "givenCardsNumber": 10,
      "freeCardsNumber": 10,
      "cardCost": 1,
      "currency": "USD"
    }
  ]
}
```


Error Codes

Tables below contain the error codes that should be returned by PROVIDER in the response.

Shared exceptions:

Code	Description
0	"OK"
100	Internal Service Error
5	Incorrect secureLogin or/and securePassword
7	Required field is empty: '{0}' is required
8	Field {0} is wrong. {0} will contain a field causing an error.
9	Player with transferred nickname already exists
10	Player is not found
11	Currency code '{0}' is incorrect or unsupported

Bingo exceptions:

Code	Description
101	You cannot order cards more than max available number
102	Pre-order of cards is not allowed at this point of time for the selected game
103	This game is not available anymore. Please order cards to another game
105	The requested data is not found
106	The cards were not found according to the criteria
107	incorrect number of cards
1	Insufficient balance
109	Wallet transaction failed
110	Max number of cards are already bought for this round

Bonus service exceptions:

Code	Description
202	Request problem. Empty mandatory field '%s'. '%s' will contain a field causing an error
203	Room(s) are not supported: %s. '%s' will contain ID of a room / rooms causing an error
204	Game(s) do not support FR bonus: %s. '%s' will contain ID of a game / games causing an error

205	May be raised when Creating a Bonus. Bonus code already exists
206	May be raised when Cancelling a Bonus. Bonus is not found or already released
207	May be raised when Cancelling a Bonus. Bonus is already canceled
208	May be raised when Cancelling a Bonus. FR bonus is started to play
211	May be raised when Creating a Bonus. Expiration date limit has been exceeded
212	May be raised when Creating a Bonus. Expiration date can't be in the past
214	May be raised when Creating a Bonus. Same bonus already exists (player, rooms, expire_date etc)
215	Request problem. Type mismatch for value '%s'. '%s' will contain a value causing an error
216	Request problem. Wrong json data: '%s'. '%s' will contain a data causing an error
217	Deprecated. Validity date must be after expiration date
219	Deprecated. Room '%s' does not support %s bonus. '%s' will contain room ID and bonus ID causing an error
220	Deprecated. May be raised when Creating a Bonus. Bonus does not support selected room
221	Invalid bonus type
222	Invalid min price
223	Invalid max price
224	Minimum price should be less or equal to maximum price

17.4 Free tickets API

ActiveGameConfigs

This method returns a list of used game configurations in each Bingo room.

Request parameters

Property	Type	Description	Status
secureLogin	string	Login to access the API	Required
currency	string	Currency symbol (ISO 4217)	Optional
roomID	Number	Bingo room identifier.	Optional

Example of request

```
POST /BingoIntegration/FreeTicketsAPI/ActiveGameConfigs HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab
```

```
{
  "currency": "USD"
, "roomID": 1582
, "secureLogin": "7casino"
}
```

Response parameters

Property	Type	Description
error	number	Response status.
description	string	Response status - short description.
rooms	array of type 'Rooms'	List of Bingo rooms with details

Structure 'Rooms'

Property	Type	Description
roomName	String	Bingo room name
roomID	Number	Bingo room identifier.
standardFT	Boolean	True or false. True - if standard free tickets can be used to play in this room
monetaryFT	Boolean	True or false. True - if monetary free tickets can be used to play in this room
gameConfigs	array of type ' gameConfigs '	List of game configurations for bingo rounds

Structure 'gameConfigs'

Property	Type	Description
gameConfigID	String	Unique game configuration ID within the Pragmatic play.
gameName	String	Name of Bingo game.
cost	Number	Cost for a ticket or cost for a pack (depending on the settings)

type	String	Defines purchase of tickets type (by one or by fixed pack)
prize	Number	Initial game prize pot for all parts in platform currency
x2goPrizes	Array of the type 'x2goPrizes'	Arrays of XtoGo prizes
jackpot	Boolean	True or false. True If the jackpot is included in this game
rollOnPrizes	Array of the type 'rollOnPrizes'	Array of Roll on prizes
prizeType	string	FRB – for FRB prizes, this value present only when special prizes
betPerLine	Number	Value of money for one line in casino games for player currency. This value present only when FRB prizes

Structure 'rollOnPrizes'

Property	Type	Description
rollOn	Number	Level of Roll on prize
prize	Number	Pot for each Roll on prize

Structure 'x2goPrizes'

Property	Type	Description
x2go	Number	Level of XtoGO prize
prize	Number	Pot for each XtoGo prize

NOTE: If currency is not sent in request, in response there will be platform currency

Example of response

```
{
  "error": "0",
  "description": "OK",
  "rooms": [
    {
      "gameConfigs": [
        {
          "gameConfigID": "1522-6505",
          "gameName": "uk fixed 75",
          "cost": 12.58,
          "type": "One ticket",
          "prize": 125.8,
          "x2goPrizes": [],
          "jackpot": false
        },
        {
          "gameConfigID": "1522-6563",
          "gameName": "free_game_uk",
          "cost": 0.0,
          "type": "One ticket",
          "prize": 30.0,
          "x2goPrizes": [],
          "jackpot": false
        },
        {
          "gameConfigID": "1522-6623",
          "gameName": "very low prize",
          "cost": 1.25,
          "type": "One ticket",
          "prize": 0.0,
          "x2goPrizes": [],
          "jackpot": true
        },
        {
          "gameConfigID": "1522-6624",
          "gameName": "very low",
          "cost": 0.13,
          "type": "One ticket",
          "prize": 0.0,
          "x2goPrizes": [
            {
              "x2go": 1,
              "prize": 12.5
            }
          ],
          "jackpot": true
        },
        {
          "gameConfigID": "1522-6513",
          "gameName": "uk mixed",
          "cost": 12.58,
          "type": "One ticket",
          "prize": 2516.0,
          "x2goPrizes": [],
          "jackpot": false
        }
      ],
      "roomName": "UK",
      "roomID": 1522,
      "standardFT": true,
      "monetaryFT": false
    }
  ]
}
```

CreateFreeTickets

The following method provides the possibility to award players with free tickets.

Request parameters

Property	Type	Description	Status
secureLogin	string	Login to access the API	Required
playerID	string	Unique player identifier	Required
freeTicketsType	string	Defines the type of free tickets <ul style="list-style-type: none">"STANDARD" (Free tickets that don't have cost and don't participate in prize contribution)"MONETARY" (Free tickets that don't have influence on player's balance but contribute and consider as normal ticket for reports)	Required
roomIDList	Array	List of symbolic unique rooms identifiers in scheduler	Required
freeTickets	Number	Number of free tickets awarded.	Required
minCostTicket	Number	Min ticket price in the game where Free tickets can be used (in platform currency)	Optional
maxCostTicket	Number	Max ticket price in the game where Free tickets can be used (in platform currency)	Optional
bonusCode	string	Unique bonus identifier within the Casino Operator system.	Required
expirationDate	Number	Date and time of free rounds expiration. (timestamp in milliseconds)	Optional

Example of request

```
POST /BingoIntegration/FreeTicketsAPI/CreateFreeTickets/ HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab
```

```
{"secureLogin":"7casino","playerID":"165f69774e61b","freeTicketsType":"STANDARD","roomIDList":[1682,1282],"freeTickets":10,"minCostTicket":0.0,"maxCostTicket":100.0,"bonusCode":"BC1587849079708","expirationDate":"1587849079708"}
```

Response parameters

Property	Type	Description
error	number	Response status.
description	string	Response status - short description
providerPlayerID	number	Player identifier within the Pragmatic Play system.

Example of response

```
{"error":0,"description":"OK","providerPlayerID":453816}
```

CancelFreeTickets

This method provides the possibility to cancel free tickets.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API	Required
bonusCode	String	Unique bonus id within the Casino Operator system.	Required

Example of request

```
POST /BingoIntegration/FreeTicketsAPI/CancelFreeTickets/ HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab
```

```
{"secureLogin": "7casino", "bonusCode": "BC1587849079708"}
```

Response parameters

Property	Type	Description
error	number	Response status.
description	string	Response status - short description
status	string	Possible values - Cancelled
bonusCode	string	Unique bonus id within the Casino Operator system.

Example of response

```
{"error": 0, "description": "OK", "status": "Cancelled", "bonusCode": "BC1587849079708"}
```

ActiveFreeTickets

This method returns free ticket info per player.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API	Required
roomID	Number	Bingo room identifier.	Optional
playerID	String	Unique player identifier.	Required

Example of request

```
POST /BingoIntegration/FreeTicketsAPI/ActiveFreeTickets/ HTTP/1.1
Host: api-bingo.prerelease-env.biz
Content-Type: application/json
x-hash: 75b4923eef6667b9833dd2fa6f3fcaab
```

```
{"secureLogin":"7casino","playerID":"165f69774e61b","roomID":1111}
```

Response parameters

Property	Type	Description
error	number	Response status.
description	string	Response status - short description
bonuses	Array of the `Bonuses`	List of Free tickets owned by the player along with their statuses

Structure 'Bonuses'

Property	Type	Description
bonusCode	String	Unique bonus id within the Casino Operator system.
roomIDList	String	Room identifiers within Pragmatic play. Represent all rooms which are connected to this bonus code
status	String	Next types possible: <ul style="list-style-type: none">Active (not all Free tickets are wagered)Pending (bonus is waiting to be claimed by player)
freeTicketsType	String	Types: <ul style="list-style-type: none">"STANDARD" (Free tickets that don't have cost and don't participate in prize contribution)"MONETARY" (Free tickets that don't have influence on player's balance but contribute and consider as normal ticket for reports)
expirationDate	Number	Date when free tickets become unavailable. Timestamp in milliseconds

activeFreeTickets	Number	Available Free tickets that can be used by the player
playedFreeTickets	Number	Free tickets that were used by the player
awardingDate	Number	Date when bonus was created in PP system. Timestamp in milliseconds
minCostTicket	Number	Min ticket price in the game where Free ticket can be used (in platform currency)
maxCostTicket	Number	Max ticket price in the game where Free ticket can be used (in platform currency)

Example of response

```
{
  "error": 0,
  "description": "OK",
  "bonuses": [
    {
      "roomIDList": "1262",
      "bonusCode": "BC23",
      "expirationDate": 1587640451179,
      "freeTicketsType": "STANDARD",
      "status": "P",
      "activeFreeTickets": 5,
      "playedFreeTickets": 0,
      "awardingDate": 1586765653150,
      "minCostTicket": 0.0,
      "maxCostTicket": 2.0
    },
    {
      "roomIDList": "1576",
      "bonusCode": "BC1587629820144",
      "expirationDate": 1587640620144,
      "freeTicketsType": "MONETARY",
      "status": "P",
      "activeFreeTickets": 5,
      "playedFreeTickets": 0,
      "awardingDate": 1586765822146,
      "minCostTicket": 0.0,
      "maxCostTicket": 2.0
    }
  ]
}
```


XVIII. Live Casino Dynamic Games Access

18.1 DGA Overview

DGA stands for Dynamic Games Access.

DGA is a data transfer system that will provide our operators with game table data to populate thumbnails on their website. Functionality will use HTML Websockets protocol to transfer the data to the operator's website in real time. Data will be consumed in JSON format.

18.2 WebSocket data feed

Pragmaticplay will provide the operator with a JavaScript API (.js file) to handle the HTML Websocket protocol. Using this API an operator will be able to make a successful subscribe to the websockets and also have functions available to listen and consume the incoming data. Doing this will allow an operator a very easy way of managing the data on their website.

18.3 Data feeds

The type of information that will be delivered to an operator through the DGA system will be the following:

- Table Name, language specific table info & image.
- Table Limits
- Table Open/Closed
- NewTable(true/false) (if new table is created and in less than 30 days, will be passed newTable:true , otherwise - newTable:false)
- Dealer Name
- Roulette
 - Amount of players on the table.
 - Hot and cold numbers.
 - Live Results
 - Last 10-20 numbers spun
 - Updated after each spin.
- Baccarat
 - Shuffle notifications.
 - Live game stats and results
 - Updated after every hand.
 - baccaratShoeSummary (gives the total games completed along with that no of games player ,banker won, tied games and bankpair and playerpair)
- Blackjack
 - Seats taken/seats open.
 - Bj game results
 - BJ game results after each game.
- SicBO
 - last 20 results (each die info and sum)
 - Statistics
 - Total seated players
 - Mega Multipliers

18.4 JavascriptAPI

Using PP's javascript API the operator will be able to handle the DGA data much easier. Depending on the chosen push or pull functionality, the .js file will include functions to subscribe, listen and consume data. The API will include the source code file(.js) and also how to use the functions provided

Latest js file can be found on:

<https://dga.pragmaticplaylive.net/dgaAPI.js>

Example of the functions contained inside the API

To start consuming the game information from Pragmatic Play's tables you will need to follow a few steps:

Step1

`dga.connect('server')`

Server could be set to live or staging URL. Pragmatic Play will provide this.

1 server connection.

`dga.available('casinoID')`

Request will return all tables available for selected casino. Pragmatic Play will provide the relevant casinoID once integration of DGA starts.

`dga.subscribe('casinoID', 'tableID', 'currency');`

Multiple subscribe methods. **1 subscribe method for each table you would like to get info from.**

Connect to a table to receive table information. tableIDs will be provided.

Currency is optional. When empty EUR values will be sent. Please consult Pragmatic Play for supported currencies.

`dga.onConnect()`

When `dga.connect('server')` has successfully connected to DGA server, then the `dga.onConnect()` method can be invoked. This can then be used to contain all the subscribe methods.

Step 2

`dga.onMessage(data)`

To consume data. Data will be contained as an object for ease of use. Override this method to start using the response information.

Please see Data format & object example on the next page. Every time the table info updates a new object will be send with the updated information to be consumed

Step 3

`dga.disconnect()`

disconnecting from service.

*when player leaves page a socket will normally auto disconnect as well.

18.5 Data Format

Data will be send to the operator using JSON format. As part of the API provided JSON format will be consumed as an object (image on the right) for ease of use.

Example of Json string for BLACKJACK.

```
{
  "availableSeats":3,
  "seat1":false,
  "seat2":false,
  "seat3":true,
  "seat4":true,
  "seat5":true,
  "seat6":true,
  "seat7":false,
  "resultSeat2":{
    "time":"Jan 10, 2017 10:09:17 AM",
    "result":"BLACKJACK"
  },
  "resultSeat3":{
    "time":"Jan 10, 2017 10:09:17 AM",
    "result":"WIN"
  },
  "resultSeat4":{
    "time":"Jan 10, 2017 10:09:17 AM",
    "result":"LOSE"
  },
  "resultSeat5":{
    "time":"Jan 10, 2017 10:09:17 AM",
    "result":"BUST"
  },
  "resultSeat6":{
    "time":"Jan 10, 2017 10:09:17 AM",
    "result":"PUSH"
  },
  "tableId":"301",
  "tableType":"BLACKJACK",
  "tableName":"TableName",
  "newTable": "false",
  "languageSpecificTableInfo":"en",
  "tableImage":"https://client.pragmaticplaylive.net/desktop/assets/snaps/sm9wsky7hh3ynx98/poster.jpg",
  "tableLimits":{
    "minBet":5,
    "maxBet":100,
    "maxPlayers":99
  },
  "dealer":{
    "name":"Aga",
    "id":""
  },
  "tableOpen":true,
  "tableTimes":"01:00-11:00 GMT"
}
```

Example of Json string for BACCARAT.

```
{
  "shuffle":true,
  "gameResult":[
    {
      "time":"Jan 10, 2017 11:12:15 AM",
      "player":2,
      "banker":4,
      "winner":"BANKER"
    }, {
      "time":"Jan 10, 2017 11:12:15 AM",
      "player":5,
      "banker":1,
      "winner":"PLAYER"
    }, {
      "time":"Jan 10, 2017 11:12:15 AM",
      "player":9,
      "banker":9,
      "winner":"TIE"
    }, ..... To a total of 20 results.
  ],
  "totalSeatedPlayers":22,
  "bankerCards": [{type: "CARD_10", sute: "HEARTS"}, {type: "CARD_A", sute: "DIAMONDS"},...],
  "playerCards": [{type: "CARD_9", sute: "SPADES"}, {type: "CARD_5", sute: "CLUBS"}, {type: "CARD_8", sute: "CLUBS"}],
  "statistics": "[[\"PNO\",\"---\",\"---\",\"---\",\"---\",\"---\"],[\"BBO\",\"BNO\",\"---\",\"---\",\"---\",\"---\"],[\"PNO\",\"PPO\",\"---\",\"---\",\"---\",\"---\"],\n\"][\"BNO\",\"BPO\",\"BNO\",\"BPO\",\"BPO\",\"---\"],[\"PBO\",\"PNO\",\"---\",\"---\",\"---\",\"---\"]]"
  "baccaratShoeSummary": {totalGames: "49", bankerWinCounter: "28",playerWinCounter: "17", tieCounter: "4",bankerPairCounter:"4",playerPairCounter: "11"}
  "tableId":"601",
  "tableType":"BACCARAT",
  "tableName":"TableName",
  "newTable": "false",
  "languageSpecificTableInfo":"en",
  "tableImage":"https://client.pragmaticplaylive.net/desktop/assets/snaps/q83swlvmxue0q5/poster.jpg",
  "tableLimits":{
    "minBet":5,
    "maxBet":100,
    "maxPlayers":99
  },
  "dealer":{
    "name":"Aga",
    "id":""
  },
  "tableOpen":true,
  "tableTimes":"01:00-11:00 GMT"
}
```

Example of Json string for ROULETTE.

```
{
  "coldNumbers": [21,15,11,33,22],
  "dealer": {
    "id": "",
    "name": "Aga"
  },
  "hotNumbers": [5,22,36,1,17],
  "languageSpecificTableInfo": "en",
  "last20Results": [
    {
      "result": 2,
      "time": "Jan 10, 2017 10:09:17 AM"
    },
    {
      "result": 3,
      "time": "Jan 10, 2017 10:09:17 AM"
    }
  ],
  "newTable": "false",
  "tableId": "201",
  "tableImage": "https://client.pragmaticplaylive.net/desktop/assets/snaps/5kvxlw4c1qm3xcyn/poster.jpg",
  "tableLimits": {
    "maxBet": 100,
    "maxPlayers": 99,
    "minBet": 5
  },
  "tableName": "TableName",
  "tableOpen": true,
  "tableTimes": "01:00-11:00 GMT",
  "tableType": "ROULETTE",
  "totalSeatedPlayers": 25
}
```

- Example of Json string for SicBO

```
{
  "betTimer": 1,
  "betsOpened": false,
  "currency": "EUR",
  "dealer": {
    "id": "lx0yiwofvqdiot1p",
    "name": "Natalya"
  },
  "languageSpecificTableInfo": "",
  "last20Results": [
    {
      "die1": 1,      "die2": 2,      "die3": 4,      "time": "Oct 12, 2020 8:30:25 AM",      "totalSum": 7
    },
    .....
    {
      "die1": 1,      "die2": 4,      "die3": 6,      "time": "Oct 12, 2020 8:19:41 AM",      "totalSum": 11
    }
  ],
  "newTable": false,
  "shuffle": false,
  "sicBoMultipliers": [
    {
      "multiplier": "20.0",      "number": "117"
    },
    .....
    {
      "multiplier": "5.0",      "number": "167"
    }
  ],
  "statistics":
  "[[\"3:4:5\\\", \"2:2:6\\\", \"4:6:6\\\", \"4:5:6\\\", \"1:2:6\\\", \"3:4:5\\\"], ..., [\"1:2:3\\\", \"2:2:5\\\", \"2:5:6\\\", \"3:4:6\\\", \"3:4:5\\\", \"2:6:6\\\"], [\"1:1:6\\\", \"4:5:5\\\", \"2:4:5\\\", \"1:2:4\\\", null, null]]\",
  \"tableId\": \"701\",
  \"tableImage\":
  \"https://client.pragmaticplaylive.net/desktop//assets/snaps/lc419kkmr2sxfpbk/poster.jpg?v0.31138094468502064\",
  \"tableLimits\": {
    \"maxBet\": 5000.0,
    \"maxPlayers\": 500,
    \"minBet\": 0.1,
    \"ranges\": [
      0.1,
      0.2,
      0.25,
      0.4,
      0.5,
      1.0
    ]
  },
  \"tableName\": \"Mega Sic Bo\",
  \"tableOpen\": true,
  \"tableType\": \"SicBO\",
  \"totalSeatedPlayers\": 83
}]}
```

XIX. Games JavaScript API

This section describes Pragmatic Games JS API which can be used by operators to take different events like spin started, ended, entering to bonus game or free spins directly from the game using JavaScript communication. Starting the game launch, Pragmatic games send different events to the topmost window, so they can be caught and used for different purposes on operator side – to attract players' attention, etc.

19.1 Game Events Interception

During lifetime, starting the moment when game is loaded into iframe, game sends different events based on user behavior to the topmost window. So that, sent events can be intercepted on operator side and used further.

The example of the code which can intercept these events looks like this:

```
XD.receiveMessage(handler, ["http://", $Host].join(""));
```

```
XD.receiveMessage(handler, ["https://", $Host].join(""));
```

Where handler - js custom function for handling the event

XD - the postMessage library

All events are sent in the following format:

```
Object {  
  name:<name of the event>  
  data:<additional data for this event>  
};
```

Example:

```
Object {  
  name: "bonusGameStarted"  
};
```

19.2 Events

Currently all Pragmatic HTML5 games send events described in the table below.

Event Name	Description	Data
spinStarted	Player clicks <i>Spin</i> button and reels started spinning	No
spinEnded	Reels stopped	No
resultShown	Game round result is shown to player and all the animation related to the round is completed	
gameRoundStarted	Player clicks <i>Spin</i>	No
gameRoundEnded	Game round is finished (it comes before resultShown event)	No
bonusGameStarted	Bonus game opens for player	No
bonusGameEnded	Player played bonus game, saw result and returned to the main game	No
freeSpinStarted	Free spins started for player	No
freeSpinEnded	Free spins are over, player saw result of the Free Spins and returned back to the main game	No
gameLoadingStarted	Game loading has started	No
gameLoadingEnded	Game has finished loading and is available for playing	No
balanceTooLow	Player no longer has enough balance to spin with current selected bet	No
notifyCloseContainer	Game container is closing because of window closure, redirect or refreshing	No
gameQuit	For home/lobby/close/leave	No
openCashier	For insufficient funds message/button	No

19.3 Triggers

Stop Autoplay

There is option to stop Auto Play for our games from client side.

It can be done by sending for iFrame with a game following:

```
postMessage({'type': "Tilt"}, '*');
```

Update balance

There is option to update balance in cases, where player's balance can be changed outside of game (for example when player deposited or playing in another game in parallel)

It can be done by sending for iFrame with a game following messages:

```
postMessage("updateBalance");
```

```
postMessage({type: "RefreshBalance"});
```


19.4 Live Casino Games Full Screen JS

Full Screen in live games allows to use total set of functionality in viewport, and ensures the game container stick without any scrolls and scaling, to give users a good interaction

In order to allow fullscreen mode, when Live Casino Game is wrapped in iframe, Operator should embed in Object (container, iframe) with game client, following JS script:

```
<script src="https://client.pragmaticplaylive.net/desktop/assets/api/fullscreenApi.js" ></script>
```