



Integration with casino games *API specification*

Latest update: February, 2019

API version 3.85

Table of Contents

- Version Control 6
- I. API Overview 10
 - 1.1 Seamless Wallet API 10
 - 1.2 Balance Transfer API 11
 - 1.3 Integration API 11
 - 1.4 Data feeds 11
 - 1.5 Player ID..... 12
 - Seamless Wallet (Authenticate method) 12
 - Balance Transfer (StartGame method) 12
 - 1.6 Play session 12
 - 1.7 Transaction reference 12
- II. Integration API 13
 - 2.1 GetCasinoGames 13
 - 2.1.1 Game icons 15
 - 2.2 DownloadStatistics 16
 - 2.3 CloseSession..... 17
 - 2.4 CancelRound 18
- III. Seamless Wallet API 19
 - 3.1 Game opening 19
 - 3.2 Hash calculation 20
 - 3.3 Data types 20
 - 3.4 Authenticate 22
 - 3.5 Authenticate2 24
 - 3.6 Balance 25
 - 3.7 Bet 26
 - 3.8 Result 28
 - 3.9 BonusWin..... 30
 - 3.10 JackpotWin..... 32
 - 3.11 EndRound 34
 - 3.12 Refund 36
 - 3.13 Withdraw 38
 - 3.14 GetBalancePerGame 39
 - 3.15 PromoWin 40
 - 3.16 Error codes 42
- IV. Balance Transfer API 43

4.1 Game opening	43
4.2 Hash calculation	43
4.3 Data types	44
4.4 CreatePlayer.....	45
4.5 Transfer	46
4.6 GetTransferStatus	47
4.7 GetBalance	48
4.8 StartGame	49
4.9 TerminateSession	50
4.10 Error codes	50
V. Simple Free Rounds API	51
5.1 Hash calculation	51
5.2 Data types	51
5.3 CreateFRB	52
5.4 CancelFRB	53
5.5 GetPlayersFRB	54
5.6 Error codes	56
VI. Variable Free Rounds API	57
6.1 Hash calculation	57
6.2 Data types	58
6.3 Create Free Rounds	58
6.4 Cancel Free Rounds	60
6.5 Add players	61
6.6 Remove players	62
6.7 Error codes	63
VII. Game History API	64
7.1 Hash calculation	64
7.2 Data types	65
7.3 GetPlayedGames	66
7.4 GetGameRounds	67
7.5 OpenHistory	69
7.6 GetRoundStatus	70
7.7 OpenHistoryExtended	71
VIII. Data feeds and reports	72
Time points	72
Using timepoints for past periods.....	72
Data Feed URL	73

Data formats.....	73
8.1 Game Rounds	74
8.2 In-game transactions	76
8.3 Failed Transactions	78
8.4 Active Jackpots	79
8.5 Jackpot Winnings	80
8.6 Daily Totals.....	81
8.7 Player's Incomplete Rounds	82
IX. Tournament API	83
9.1 Tournament Winners	83
X. Diagrams	85
10.1 Flow diagram of opening casino game for Seamless Wallet API.....	85
10.2 Flow diagram of playing casino game Seamless Wallet API.....	86
XI. Reconciliation	88
11.1 Reconciliation process	88
Bet transaction	88
Refund transaction	88
Winning transactions	88
End round	88
11.2 Reconciliation mechanism	88
XII. Reality Check	89
12.1 Server side implementation	89
Reality Check Warning	89
Reality Check Response.....	90
12.2 Client side implementation	91
XIII. Custom messages	92
13.1 Custom messages syntax	92
13.2 Action types	93
13.3 Link types	93
13.4 Custom messages API	94
XIV. Responsible gaming - Sweden	95
14.1 Login time	95
14.2 Links and logos to self-assessment tests	95
14.3 Set player's country	95
14.4 Displaying of totals	95
XV. Integration with demo games	96
XVI. Bingo Integration API	97

Bingo Game API	97
Definitions	97
16.1 Bingo room Launching	97
Room opening	97
Mini slots	100
16.3 Bingo integration API for OPERATOR side.....	101
16.4 Bingo integration API for PROVIDER side	101
RoomList	101
AvailableGames	103
PreorderCards	105
PreorderedReport	107
CancelPreorderedCards	109
JackpotWinners	110
ChangeNickname	111
Bingo Player History	112
Bingo Round History	114
Error Codes	116
16.5 Bingo Free rounds API	117
Simple Free Rounds API	117

Version Control

Date	Author	Version	Changes description
28-Nov-14	Iryna	3.0	<ol style="list-style-type: none"> Added GetCasinoGames method on the Pragmatic Play side. Fields "secureLogin" and "securePassword" added to all methods on the Operator side except GetVersion.
9-Dec-14	Iryna	3.1	API will support player id within Operator system for Option 2 (Authenticate method + player auto-registration) and within Pragmatic Play system for Option 1 (StartGame method, compatible with the v2).
29-Jan-15	Iryna	3.2	If player is not authenticated, 404 error will not appear anymore.
19-Mar-15	Iryna	3.3	<ol style="list-style-type: none"> Added Simple HTTP service chapter. Added Error code chapter.
26-Mar-15	Iryna	3.4	Parameter "platform" for CasinoGame type will be detailed by "iOS" and "Android" values.
7-Apr-15	Iryna	3.5	Added timestamp in GMT to Bet and BetResult methods. Added round details field GMT to Bet and BetResult methods.
14-Apr-15	Iryna	3.6	Added Data Feeds section.
15-May-15	Iryna	3.7	Added new Authenticate2 method (only for casino client integration).
2-Jun-15	Iryna	3.7	Added Game Opening chapter (just for information, no changes in the API).
9-Jul-15	Iryna Sh.	3.8	Added Free Bonus Round API section. Added bonusWin method to the Wallet API on Operator side.
16-July-15	Ira K	3.8	Added new UpdateBalance method to Integration API
16-Jul-15	Iryna Sh.	3.9	Wallet API updates: Added "platform" parameter to Bet, BetResult and Refund methods. Added "bonusCode" parameter to Bet, BetResult methods.
28-Jul-15	Iryna Sh.	3.10	Added Wallet API -> EndRound method.
5-Aug-15	Iryna Sh.	3.10	Added error codes for FR Bonus API
13-Aug-15	Iryna Sh.	3.11	UserId type changed to String instead of Long (http API only).
16-Sep-15	Iryna Sh.	3.12	Free rounds bonus API -> Added GetPlayersFRB method (available only for http service)
21-Sep-15	Iryna Sh.	3.13	Added aspectRatio and demoGameAvailable fields to Integration API -> GetCasinoGames method.
21-Sep-15	Iryna Sh.	3.14	Added new History API (HTTP service).
29-Sep-15	Iryna Sh.	3.15	Updated the list of error codes for Free Rounds Bonus Http service. Expiration date type has been replaced by Long (timestamp in sec.)
30-Sep-15	Iryna Sh.	3.15	FR Bonus Http service -> Expiration date is mandatory, 30 days max. Added originalSessionID (SOAP), originalRoundID (http service) to Bet, Result and End round methods.
13-Oct-15	Iryna Sh.	3.16	Added Integration http service -> Download statistics method.

16-Nov-15	Iryna Sh.	3.17	Added DOWNLOAD platform value to GetCasinoGames method. Values are comma separated.
10-Dec-15	Iryna Sh.	3.18	Added Integration http service -> GetCasinoGames method.
11-Dec-15	Iryna Sh.	3.19	Added Wallet API -> JackpotWin method to both SOAP and http services
15-Dec-15	Iryna Sh.	3.20	Added Data feeds -> Actual Jackpots, Jackpot winnings.
26-Jan-16	Iryna Sh.	3.21	Optional "amount", "gameID", "roundID", "timestamp", "roundDetails", "bonusCode", "originalRoundID" fields have been added to Wallet Http Service -> Refund.
17-Mar-16	Iryna Sh.	3.30	Simple Integration http API -> Added methods for single wallet: CreatePlayer, GetBalance, Transfer, StartGame and TerminateSession method for kicking player out of the games.
22-Apr-16	Iryna Sh.	3.31	Wallet Simple http service -> Withdraw method has been added for downloadable client.
2-Jun-16	Iryna Sh.	3.32	Error codes: 50 error code has been added for regulated markets.
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 Balance Transfer error codes Added 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	<i>Add Error 8 to Balance Transfer API Error codes</i>
12-May-17	Iryna Sh.	3.56	<i>Statuses “CompleteInProcess” and “CancelInProcess” have been added to GetRoundStatus method in History API.</i>
19-May-17	Alex N.	3.57	<i>userId parameter is case-sensitive. Reality check response: request/response examples.</i>
23-Jun-17	Iryna Sh.	3.58	<i>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.</i>
11-Jul-17	Alex B.	3.59+zh	<i>Optional token parameter added into Seamless Wallet API methods</i>
19-Jul-17	Iryna Sh.	3.60	<i>Reality check chapter has been divided into two parts – server side and client side implementations.</i>
9-Aug-17	Iryna Sh.	3.61+zh	<i>Added new Balance Transfer API -> GetTransferStatus method.</i>
10-Aug-17	Iryna Sh.	3.62	<i>Added country and jurisdiction parameters to the Seamless Wallet API -> Authenticate response.</i>
18-Aug-17	Iryna Sh.	3.63+zh	<i>Added new Data Feeds API -> Game Rounds method.</i>
22-Aug-17	Iryna Sh.	3.64	<i>Added new Integration API -> CancelRound method.</i>
23-Aug-17	Iryna Sh.	3.65	<i>Added new Data Feeds -> Daily Totals method.</i>
31-Aug-17	Iryna Sh.	3.66	<i>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.</i>
6-Sep-17	Alex F.	3.67	<i>Updated send Reconciliation retry call for Seamless Wallet API Error Codes</i>
13-Nov-17	Alex F.	3.68	<i>Added demo link for Asia Region</i>
16-Nov-17	Iryna Sh.	3.69	<i>Added optional jackpotContribution and jackpotId fields to Seamless Wallet API -> Bet. Configurable per brand.</i>
7-Dec-17	Iryna Sh.	3.70	<i>Free Rounds can be cancelled regardless of whether the player started to play them or not.</i>
24-Jan-18	Iryna Sh.	3.71	<i>Tournament API has been added.</i>
30-Jan-18	Iryna Sh.	3.72	<i>Date-time format was added to the Tournament API specs and examples.</i>
27-Feb-18	Iryna Sh.	3.73	<i>New betLimit parameter has been added to Seamless Wallet API -> Authenticate response, gameId is added to Authenticate request.</i>
5-Mar-18	Iryna Sh.	3.74	<i>Error code 310 has been added to Seamless Wallet API.</i>
23-Mar-18	Alex Ch.	3.75	<i>OpenHistoryExtended has been added.</i>
26-Apr-18	Alex Ch.	3.75	<i>Data Feeds -> Daily Totals. Added "totalFreeSpinRounds" response parameter</i>
4-May-18	Alex Ch.	3.75	<i>Added: ipAddress to 3.3 Data Types, 3.4 Authenticate, 3.7 Bet Added: usedPromo to 3.3 Data Types</i>
17-May-18	Iryna Sh.	3.76	<i>Method Result: added parameters for Mystery Race winnings got by player during game round.</i>
24-May-18	Iryna Sh.	3.77	<i>Added Custom Messages chapter.</i>

12-Jul-18	Iryna Sh.	3.78	<i>Changes in Custom Messages:</i> <ol 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 Bul.	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

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 “10.1 Flow diagram of opening casino game ”	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.	Optional ²
JackpotWin	Using this method a Pragmatic Play system will notify Casino Operator about Jackpot winning.	Optional ³
EndRound	Notifies the Casino Operator that the game round is over, for transactions finalization purpose.	Optional
Withdraw ⁴	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 Is required in case of Free Round Bonuses API is implemented

³ Is required in case of Jackpot functionality is implemented

⁴ 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 ^{5*}	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

⁵ 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, wins 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
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
gameID	String(20)	Symbolic unique identifier of the game provided by Pragmatic Play. Example: <i>vs7monkeys</i> , <i>vs50amt</i> , <i>rla</i> , <i>wra</i> , <i>bjb</i>
gameName	String(100)	Name of the game. Example: <i>"Lucky Dragons"</i> , <i>"Hot Safari"</i>
gameTypeID	String(20)	Symbolic unique identifier of the game. Example: <i>"vs"</i> , <i>"rl"</i> , <i>"vp"</i> .
typeDescription	String(100)	Short description of the game type. Example: <i>"Video Slot"</i> , <i>"Roulette"</i> , <i>"Video Poker"</i> .

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 gamesflash – Flash (only for desktop) <i>Example: “flash,html5”, “html5”</i>
technologyID	String(10)	Technologies in which the game is available, comma separated. Possible values are: <ul style="list-style-type: none">H5 - HTML5 gamesF – Flash (only for desktop) <i>Example: “H5”, “F,H5”</i>
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 devicesWEB – if game will be opened on desktop devicesDOWNLOAD – if game can be opened in downloaded gaming client
demoGameAvailable	Boolean	If true, a demo version of the game is available.
aspectRatio	String(10)	Describes the proportional relationship between the width and the height of the game. <i>Example: “4:3”, “16:9”</i>
gameIdNumeric	Long	The numeric value of gameId

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
```

Example of JSON response:

```
{
  "error": "0",
  "description": "OK",
  "gameList": [
    {
      "gameID": "vs50hercules",
      "gameName": "Hercules Son of Zeus",
      "gameTypeID": "vs",
      "typeDescription": "Video Slots",
      "technology": "html5",
      "platform": "MOBILE,DOWNLOAD,WEB",
      "demoGameAvailable": true,
      "aspectRatio": "16:9",
      "technologyID": "H5",
      "gameIdNumeric": 1477914757
    }, ...
    {
      "gameID": "vpa",
      "gameName": "Jacks or Better",
      "gameTypeID": "vp",
      "typeDescription": "Video Poker",
      "technology": "flash,html5",
      "platform": "MOBILE,DOWNLOAD,WEB",
      "demoGameAvailable": true,
      "aspectRatio": "5:4",
      "technologyID": "F,H5",
      "gameIdNumeric": 1455872884
    }
  ]
}
```

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.
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&roundId=123456&hash=5fd31717ba4da9e78ac19730212925c2
```

Example of JSON response:

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

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

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 one-time 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;
- 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}%26technolo
gy%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.

Also game opening link can be composed explicitly, in any other convenient for Operator way.

Example of game opening link:

```
https://{game_server_domain}/playGame.do?key=token%3Drgk1r4d5b6%26symbol%3Dvs20b1%26technology%3DH5
%26platform%3D%26language%3Den%26cashierUrl%3Dhttp%3A%2F%2Fvs20b1%26lobbyUrl%3D&stylename=username
```

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 ○ GI - Gibraltar ○ RO - Romania ○ MT - Malta ○ PH - Philippines ○ LT – Lithuania ○ LV - Latvia ○ EE - Estonia ○ SE - Sweden ○ 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>
roundId	Long	Identification for the player's round.
jackpotId	Long	Identification for the jackpot.
jackpotContribution	Decimal (10, 2)	Amount of the contribution to the jackpot

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.
token	String (32)	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 Pragmatic Play 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. Example: “bonus, gamble”.
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.

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. Token can be used only once.

Request parameters

Name	Description	Status
hash	Hash code of the request.	Required
token	Token of the player.	Required
providerId	Game Pragmatic Play 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. 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

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 3.3 <i>Data types</i> section for more details.	Optional
betLimits	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: defaultBet – the default bet value minBet – the minimal individual bet amount maxBet – the maximal individual bet amount maxTotalBet – the maximum total bet amount (in the game currency) minTotalBet – the minimum total bet amount (in the game currency) <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

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: someoperatoromain.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: someoperatordomain.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 Pragmatic Play 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 Pragmatic Play 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
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
jackpotId	Id of the active jackpot to contribute. The field is optional and should be sent together with jackpotContribution. 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

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=1482429190
```

374

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 Pragmatic Play 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. <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
promoWinReference	Unique reference of this transaction. <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
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

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 Spin 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 Pragmatic Play 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	Optional
gameId	Id of the game for the last played round within Free Round Bonus	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 /bonusWin.html HTTP/1.1
Host: someoperatoromain.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 Pragmatic Play 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=585d0b2af89c56f5ed6f0d1f&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": 99815904,
  "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 Pragmatic Play id.	Required
platform	<div>The platform type (channel) on which the game is played.</div> <div>Possible values:</div> <div><ul style="list-style-type: none">“MOBILE” – mobile device“WEB” – desktop device“DOWNLOAD” – downloadable client</div> <div>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.</div>	Optional
token	<div>Token of the player from Authenticate response.</div> <div>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.</div>	Optional

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 for rollback a bet on the Casino Operator side, in order to add money back that was taken during the bet.

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

Important: If bet 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 bet transaction.	Required
providerId	Game Pragmatic Play 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. 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

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 Pragmatic Play 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=b4672931ee1d78e4022faaadf58e37db
```

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 game 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 Pragmatic Play 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	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 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
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.	Yes	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=620a4016ae32ea0efbebfelb3544571c
```

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: 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.

Code	Description
1	Internal error. Try later please.
2	Incorrect secure LOGIN and secure password combination (will be returned also if the hash code validation is failed)
6	Game is not found or is not allowed for your system
7	One or several input parameters is not set or set incorrectly.
8	Transaction already exists
17	Player not found
21	Currency code is incorrect or unsupported

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	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
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=vs20bl&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
rounds	Number of free game rounds awarded.	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/create HTTP/1.1
Host: api.prerelease-env.biz
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

secureLogin=username&bonusCode=421&startDate=1482588510&expirationDate=1482598510&rounds=10&
hash=1d319398565e48dc635c3c621475d0ac

{
  "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 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"
}
```

6.6 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 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.

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=9KngwmdYVw6YLViPVIjU9eEMFslzlALnw
toZygNWTMB7SqAAp18ERLysn7Xp26p0&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
roundId	Unique identifier of the game round.	Required
playerId	Id of the player within the Operator system.	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&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"
}
```

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.

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 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	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
timepoint	Long	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

Response parameters (CSV)

Name	Type	Description
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). 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)
startDate	DateTime	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")
endDate	DateTime	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")

status	String(1)	Status of the game round: I – In Progress (not finished yet) C – Completed
type	String(1)	Type of the game round: R – game round F – free spin triggered during the game round
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.

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=1482510800258
playerID,extPlayerID,gameID,playSessionID,parentSessionID,startDate,endDate,status,type,bet,win,cur
rency, jackpot
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
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
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
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
```

8.2 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	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
timepoint	Long	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

Response parameters (CSV)

Name	Type	Description
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).
timestamp	DateTime	Date and time when the transaction is processed on the Pragmatic Play side. (Unix epoch time in milliseconds, for example : 1470926696715)
referenceID	String(100)	Unique id for reference of this transaction on the Pragmatic Play side.
type	String(1)	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
amount	Double(15,2)	Amount of the transaction.
currency	String(3)	Currency of the transaction, 3 letter ISO code.

Example of HTTP request:

```
GET /IntegrationService/v3/DataFeeds/transactions/?
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,timestamp,referenceID,type,amount,currency
6638030,421,vs50aladdin,5111126041,1482510741000,585d5195f89c56f5ed95012e,B,2.50,USD
6638030,421,vs50aladdin,5111126041,1482510742000,585d5196f89c56f5ed950159,W,0.50,USD
6638030,421,vs50aladdin,5111126590,1482510745000,585d5199f89c56f5ed950344,B,2.50,USD
6638030,421,vs50aladdin,5111127141,1482510748000,585d519cf89c56f5ed95054e,B,2.50,USD
```

8.3 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	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
timepoint	Long	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	Type	Description
playerID	Number(15)	Player unique identifier within Pragmatic Play system.
extPlayerID	Number(15)	Player unique identifier within Casino Operator system.
gameID	String(20)	Symbolic unique identifier of the game within the Pragmatic Play system. Example: vs25queenofgold, vs50aladdin, cs5triple8gold, bjma, rla, vpa The list of available games may be obtained by Casino Operator by calling GetCasinoGames method of CasinoGameAPI HTTP service.
playSessionID	Number(15)	Id of the player's session for particular game (unique number of the game round).
referenceID	String(100)	Unique id for reference of this transaction on the Pragmatic Play side.
betAmount	Double(15,2)	Amount of the bet.
winAmount	Double(15,2)	Amount of winning in current bet.
timestamp	Long	Date and time when the transaction is processed on the Pragmatic Play side. (Unix epoch time in milliseconds, for example : 1470926696715)
roundDetails	String(50)	Additional information about the current game round, comma separated. Example: "spin,bonus"

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
```

8.4 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	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

Response parameters (JSON)

Name	Type	Description
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: N – Network jackpot, O – Jackpot for particular Operation B – Jackpot for particular casino Brand
amount	Number(15,2)	Jackpot fund for the moment of request, in USD.
games	String(255)	The list of the games participating in the Jackpot. It contains gameId (game symbols), comma separated. <i>Example: vs25safari_jp,vs30catz_jp,vs7monkeys_jp</i>

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

8.5 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	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
jackpotID	Number(15)	Unique identifier of the Jackpot within the Pragmatic Play system.	Required

Response parameters (JSON)

Name	Type	Description
jackpotID	Number(15)	Unique identifier of the Jackpot within Pragmatic Play system.
dateTime	DateTime	Date and time when the Jackpot was won.
amount	Number(15,2)	Amount of Jackpot that was won, \$USD
gameID	String(20)	Symbolic unique identifier of the game where Jackpot was won.

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"
}
```

8.6 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	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	DateTime	Start date and time of the period for which the data should be loaded	Required
endDate	DateTime	End date and time of the period for which the data should be loaded	Required

Response parameters (JSON)

Name	Type	Description
totalRounds	Integer	Total number of game rounds completed during the reporting period, including free spin rounds number
totalFreeSpinRounds	Integer	Total number of free spin rounds completed during reporting period
totalBet	Double(15,2)	Total bet amount.
totalWin	Double(15,2)	Total win amount.
totalJackpot	Double(15,2)	Total Jackpot win amount.
currency	String(3)	Currency, 3 letter ISO code.

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
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,
      "currency": "RMB"
    },
    {
      "totalRounds": 148,
      "totalFreeSpinRounds": 28,
      "totalBet": 148,
      "totalWin": 920.68,
      "totalJackpot": 166,
      "currency": "USD"
    }
  ]
}
```

8.7 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	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
playerId	String	Id of the player within the Casino Operator's system.	Required

Response parameters (JSON)

Name	Type	Description
playerId	String	Id of the player within the Casino Operator's system.
gameId	String	Id of the game.
playSessionID	Number(15)	Id of the player's session for particular game (unique number of the game round).
betAmount	Double(15,2)	Bet amount.

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": "vs20bl",
      "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.</i>
prizeCoins	Number(15)	Please note, that prizeAmount, prizeCoins, prizeCurrency should appear together. Prize amount in coins. <i>The field is optional and will appear only if there is no prizeGift field in the response.</i> Please note, that prizeAmount, prizeCoins, prizeCurrency should appear together.
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.</i> Please note, that prizeAmount, prizeCoins, prizeCurrency should appear together.

Example of HTTP request:

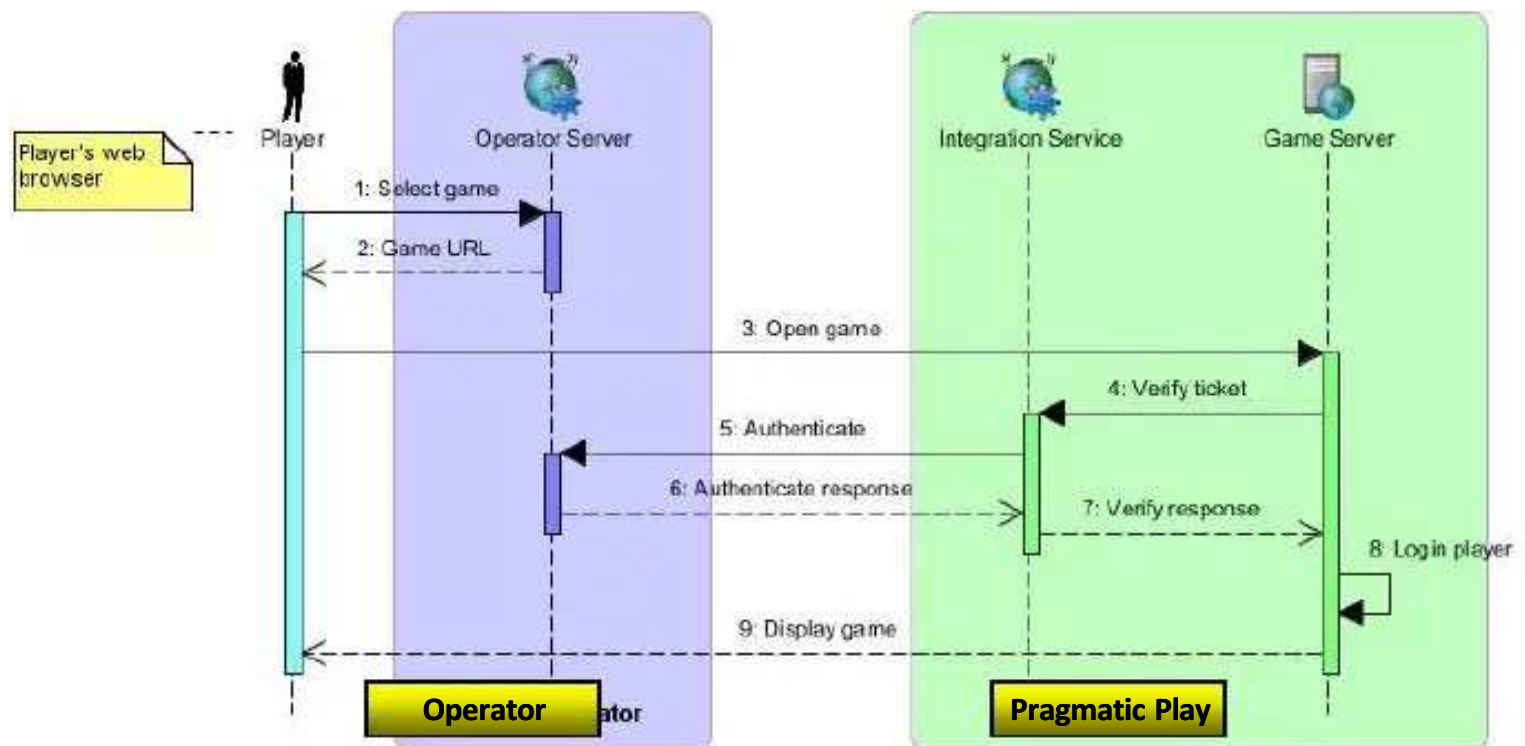
```
GET
/IntegrationService/v3/http/tournaments/winners/?login=username&password=testKey&startDate=
2018-01-20 11:00:00&endDate=2018-01-20 12:00:00 HTTP/1.1
Host: api.prerelease-env.biz
Cache-Control: no-cache
```

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. Diagrams

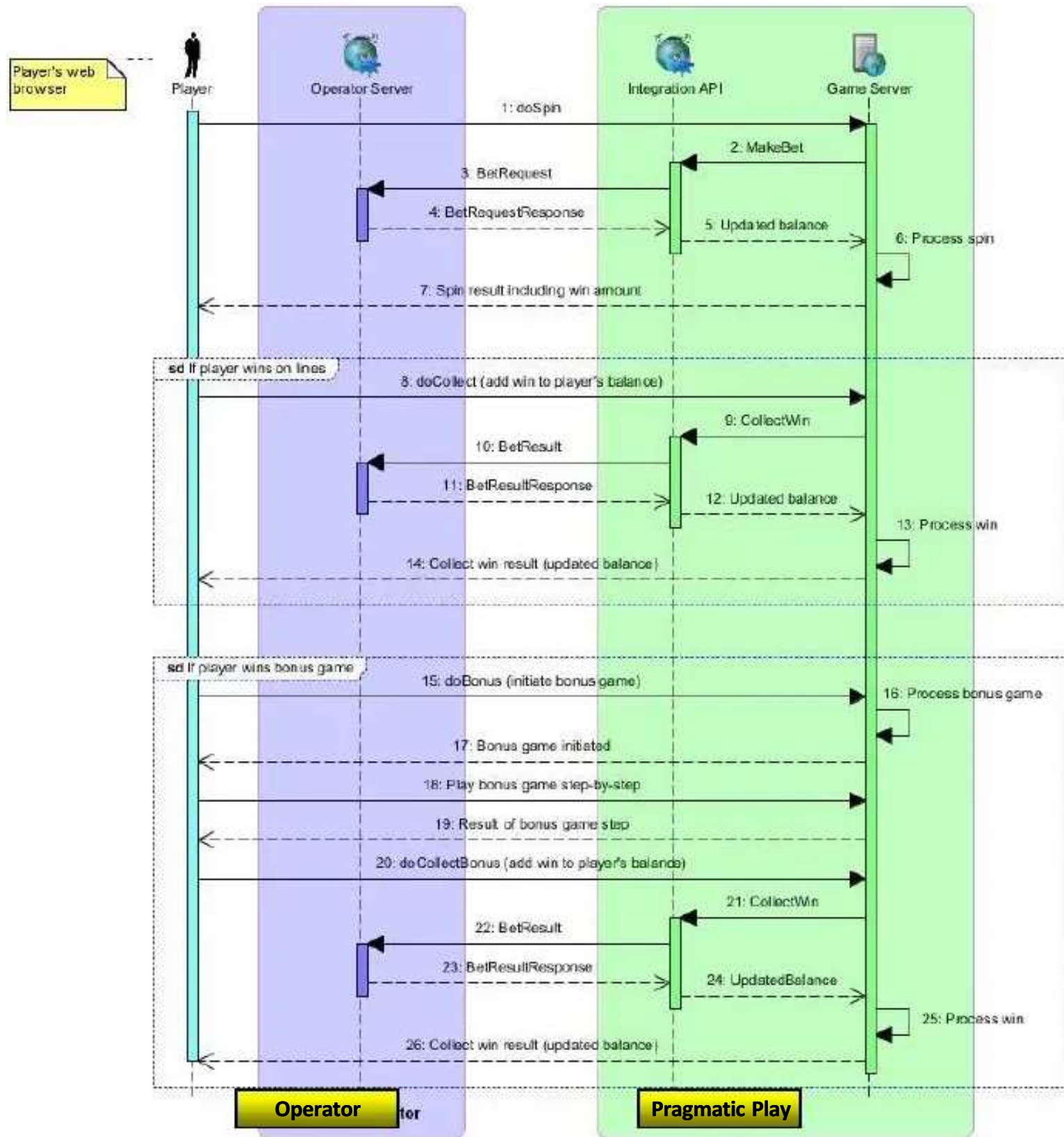
10.1 Flow diagram of opening casino game for Seamless Wallet API



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.

10.2 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.

XI. Reconciliation

11.1 Reconciliation process

If a request times out because of internet connection problems, 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 (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 on the EndRound call, so it will try to run it in the background until the Operator response will not reach Pragmatic Play.

11.2 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.

XII. Reality Check

12.1 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"
}
```

12.2 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.

XIII. 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.

13.1 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.

13.2 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.

13.3 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> .

13.4 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

XIV. Responsible gaming - Sweden

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

14.1 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.

14.2 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”

14.3 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”.

14.4 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.

XV. 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}&lobbyUrl={URL}
```

Parameter	Name	Description	Example
demo_games_domain	demo games domain	Domain configured for demo games	<i>demogamesfree.pragmaticplay.net</i> <i>demogames.pragmaticplay.net</i>
gameSymbol	game symbol	A symbol that is provided for every game that is available as demo game.	vs20bs, vs25f, bjb, kna, vpa
lang	language	The game language. If it is not passed the 'en' will be used as default. The list of available languages is described in Game Library.	de, en, es, fr, it, ja, ru, th, tr, vi, zh
cur	currency symbol	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 Game Library	USD, EUR, AUD, ZAR etc.
lobbyURL	lobby URL	Encoded URL for redirecting, if player clicks <i>Back to lobby</i> button. This URL may contain different parameters and should be started with <i>http(s)://</i> There 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>

Example of demo game opening link:

Asia region:

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

Europe, rest of the world:

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

XVI. 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.

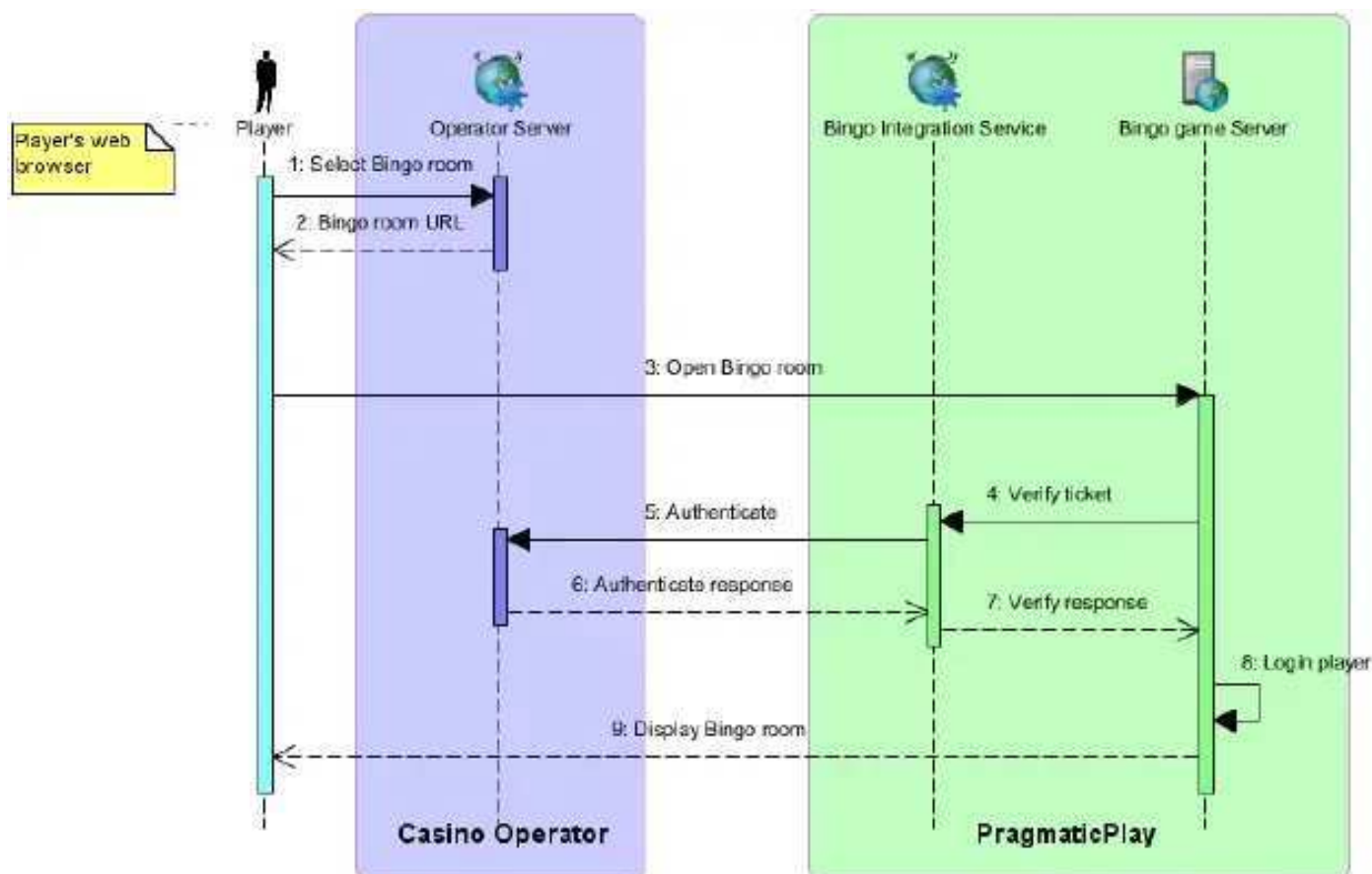
Definitions

List of definitions for the OPERATOR:

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

16.1 Bingo room Launching

Room opening



In order to retrieve URL for game launching, the Operator needs to call JS function 'BingoLib.js.gameUrl' (which is available via BingoLib.js at https://<bingo_game_server_domain>/bingohtml5/BingoLib.js) with the following parameters:

- serverURL – domain name for bingo game opening and playing;
- domain – specific setting to put OPERATOR URLs inside bingo rooms
- token – secure one-time token is generated by OPERATOR for specific player;
- room – unique identifier for bingo room;

- 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;
- commonGameMenu – URL for page with other games from OPERATOR in specific bingo room;
- gamesLobby – external html page which shows user a list of mini slots and manages to launch them ([Mini Slots](#))
- profileURL – user profile page;
- preByCardURL – page where user can make cards preorder (applies to all rooms);
- gameHistoryURL – page where user can check his game history on OPERATOR side;
- promotionURL – page which includes promotion information;
- stylename – unique identifier of the OPERATOR at PROVIDER side.
- 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
- helpURL – a link that opens the website ‘Support’ page in a new tab.
- Jurisdiction – player’s jurisdiction. Available values:
 - BS - Bahamas
 - BG - Bulgaria
 - DK - Denmark
 - UK - United Kingdom
 - IT - Italy
 - ES - Spain
 - PT – Portugal
 - 99 - Unregulated market
- isNewWindow – special flag for lobby client, used for clarification for lobby client to be opened in iframe or in a new window. Available values:
 - true – new window
 - false – iframe inside OPERATOR’s website

The function of call should look like: BingoLib.prepareUrl = function (serverURL, domain, token, room, language, cashierUrl, secureLogin).

Call of function BingoLib.gameUrl returns URL for the game launch.

Example:

<https://<serverURL>/bingo/playBingo.do?key=domain%3D<domain>%26token%3D<token>%26room%3D<roomID>%26language%3D<language>%26cashierURL%3D<cashierURL>%26lobbyURL%3D<lobbyURL>%26commonGameMenu%3D<commonGameMenuURL>%26profileURL%3D<profileURL>%26preBuyCardURL%3D<preBuyCardURL>%26gameHistoryURL%3D<gameHistoryURL>%26promotionURL%3D<promotionURL>%26rci%3D0%26rce%3D0%26rcHistoryUrl%3D<rcHistoryUrl>%26rcCloseUrl%3D<rcCloseUrl>%26readMoreURL%3D<readMoreURL>%26helpURL%3D<helpURL>%26gamesLobby%3D<gamesLobbyURL>%26responsibleGaming%3D<responsibleGamingURL>&stylename=operator stylename>

where all values in <...> are equal to BingoLib.gameUrl function call parameters. All parameters inside 'key' should be url-encoded except stylename parameter.

Mini slots

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 games list which will allow player to choose and play mini slots inside Bingo client. Structure and size of lobby page elements should be developed according to the most required device models. Iframe has size approximately equal to one third of the screen or with height equal to half of display length in mobile version. The width in desktop version is 482px and height is 328px.

Also OPERATOR needs to implement special post notification messages for Bingo web client page about actions inside iframe.

Implementation of post messages by the dispatched event:

‘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;

    // ...
}
```

The properties of the dispatched message are:

data - object passed from the other window.

origin - origin of the window that sent the message at the time ‘postMessage’ was called. This string is the concatenation of the protocol and "://", the host name if one exists, and ":" followed by a port number if a port is present and differs from the default port for the given protocol. Examples of typical origins are https://example.org (implying port 443), http://example.net (implying port 80), and http://example.com:8080. Note that this origin is not guaranteed to be the current or future origin of that window, which might have been navigated to a different location since postMessage was called.

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

source - A reference to the window object that sent the message; you can use this to establish two-way communication between two windows with different origins.

Samples which Bingo web client expects to get:

window.parent.postMessage('open-lobby', '<site domain of lobby page>') - when lobby page opened in iframe

window.parent.postMessage('open-game', '<site domain of lobby page>') – when game is launching from lobby page

window.parent.postMessage('close-game', '<site domain of lobby page>') – when game is closed page should be redirect for lobby page again and notify bingo client.

For more information use the following documentation:

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

16.3 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](#).

16.4 Bingo integration API for PROVIDER side

RoomList

This method returns list of active bingo rooms at the current moment.

Request parameters

Property	Type	Description	Status
playerID	string	Unique player identifier on operator's side	Optional
language	string	Player's language	Optional
secureLogin	string	Login to access the API	Required

Example request

```
https://{bingo API domain}/BingoIntegration/BingoGameAPI/RoomList HTTP/1.1
{
  "playerID": "playerID",
  "language": "en",
  "secureLogin": "login1526992641791"
}
```

Response parameters

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

Structure 'ActiveRoom' & Structure 'UpcomingRoom'

Property	Type	Description	Status
roomID	long	ID of the room.	Required
bingoType	integer	Bingo game type (75 or 90)	Required
roomName	string	Name of the room.	Required
cardCost	array of type 'CurrencyEntry'	Card price.	Required

gamePrize	array of type 'CurrencyEntry'	Game prize.	Required
nextGameStart	number	Starting time in UNIX timestamp in ms of the next game.	Optional
jackpotAmount	array of type 'CurrencyEntry'	Jackpot (if configured).	Required
playersBoughtCards	integer	Number of players who bought cards for current game.	Required
playersInRoom	integer	Number of players in current bingo room.	Required

Structure 'CurrencyEntry'

Property	Type	Description	Status
symbol	string	Currency symbol by ISO 4217 (USD, AUD, EUR etc)	Required
value	double	Amount of money in related parameter.	Required

Example response

```
{
  "activeRoomList": [
    {
      "bingoType": 75,
      "cardCost": [
        {
          "symbol": "AUD",
          "value": 1.34
        }
      ],
      "gamePrize": [
        {
          "symbol": "AUD",
          "value": 1.27
        }
      ],
      "jackpotAmount": [
        {
          "symbol": "AUD",
          "value": 0
        }
      ],
      "nextGameStart": 1544518740000,
      "playersBoughtCards": 0,
      "playersInRoom": 0,
      "roomID": 1182,
      "roomName": "vcb only 75 en"
    },
    {
      "description": "OK",
      "error": 0,
      "upcomingRoomList": [
        {
          "bingoType": 75,
          "cardCost": [
            {
              "symbol": "AUD",
              "value": 1.34
            }
          ],
          "gamePrize": [
            {
              "symbol": "AUD",
              "value": 201
            }
          ],
          "jackpotAmount": [
            {
              "symbol": "AUD",
              "value": 1344.02
            },
            {
              "symbol": "EUR",
              "value": 852.55
            }
          ],
          "nextGameStart": 1544518800000,
          "playersBoughtCards": 0,
          "playersInRoom": 0,
          "roomID": 1502,
          "roomName": "no loop room"
        }
      ]
    }
  ]
}
```

AvailableGames

Method returns list of rooms with games schedule (when which game will be played). Also, it will also return information about player's pre-ordered cards.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API	Required
Date	number	Date (in UNIX timestamp in ms)	Required
Hour	number	Hour in 24h format .Example: 01 means 01 AM; 16 means 04 PM.	Required
playerID	String	Unique player identifier on operator's side	Required
Language	String	Player's language by ISO 639-1	Optional

Example request

```
https://{bingo API domain}/BingoIntegration/BingoGameAPI/AvailableGames
{
  "secureLogin":"securelogin",
  "playerID":"91446987",
  "language":"en",
  "date":1544603284186
}
```

Response parameters

Property	Type	Description	Status
error	Number	Response status.	Required
description	String	Response status short description.	Required
roomList	array of type 'Room'	All rooms which are scheduled for current brand.	Required

Structure 'Room'

Property	Type	Description	Status
roomID	Number	ID of the room	Required
roomName	String	Name of the room.	Required
bingoType	Integer	Bingo game type (75 or 90)	Required
gameListByHours	array of type 'Game'	The list of games in the room at the specified hour.	Required

Structure 'Game'

Property	Type	Description	Status
gameID	Number	ID of the game. (Bingo game ID)	Required
gameName	string	Name of the game.	Required

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.	Required
maxCardsNumber	integer	Max number of cards available for purchase.	Required
cardCost	array of type 'CurrencyEntry'	Property 'cardCost': list of pairs “currency Symbol – value (double)”	Required
purchaseMode	String	C – Card / Ticket F – Fixed price S – Strips Fixed price T – Strips Ticket price	Required
players	integer	The number of players who have already preordered cards for the game.	Required
boughtCardsNumber	integer	Number of cards preordered by player.	Required
givenCardsNumber	integer	Number of free cards that was received with preordered cards.	Required
totalAmount	array of type 'CurrencyEntry'	The total amount spent by player for preordering cards.	Required

Structure 'CurrencyEntry'

Property	Type	Description	Status
symbol	string	Currency symbol by ISO 4217 (USD, AUD, EUR etc)	Required
value	double	Amount in relevant currency .	Required

Example response

```
{ "description": "OK", "error": 0, "roomList": [ { "bingoType": 75, "gameListByHours": { "0": [ { "boughtCardsNumber": 0, "cardCost": [ { "symbol": "AUD", "value": 6.08 }, { "symbol": "EUR", "value": 4.23 } ], "gameID": 5880, "gameName": "strip_fixed", "givenCardsNumber": 0, "maxCardsNumber": 2, "players": 0, "purchaseMode": "S", "serialGameIdx": 1, "totalAmount": [ { "symbol": "AUD", "value": 0 }, { "symbol": "EUR", "value": 0 } ] } ], "roomID": 1282, "roomName": "strip_fixed" } ] }
```

PreorderCards

Method allows to make cards pre-order.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API	Required
Date	Number	Date for preordering cards. UNIX time in ms	Required
Hour	Integer	Hour for preordering cards in 24h format. Example: 01 means 01 AM; 16 means 04 PM.	Required
playerID	String	Unique player identifier.	Required
nickname	string	Player's nickname.	Optional
currency	String	Player's currency.	Required
roomID	Number	ID of the room.	Required
preorderCardList	array of type 'PreorderCard'	List of pre-ordered cards for chosen games.	Required

Example request

```
https://{bingo API domain}/BingoIntegration/BingoGameAPI/PreorderCards
{
  "secureLogin":"securelogin",
  "playerID":"16243a2ec4218f2",
  "date":1544606366093,
  "currency":"EUR",
  "roomID":1442,
  "preorderCardList":
    [{"hour":10,"gameID":6000,"serialGameIdx":1,"cardNumber":20},
     {"hour":11,"gameID":6000,"serialGameIdx":1,"cardNumber":20},
     {"hour":12,"gameID":6000,"serialGameIdx":1,"cardNumber":20}]
}
```

Structure 'PreorderCard'

Property	Type	Description	Status
Error	integer	Response status.	Required
description	string	Response status short description.	Required
gameID	number	ID of the game. (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 same hour.	Required
cardNumber	integer	Number of pre-ordered cards.	Required
givenCardsNumber	integer	The number of cards given to player in addition (relevant for Buy X get Y games).	Required
Hour	integer	Hour for preordering cards in 24h format.Example: 01 means 01 AM; 16 means 04 PM.	Required

Response parameters

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

Example response

```
{
  "error": 0,
  "description": "OK",
  "preorderCardList": [
    {
      "error": 0,
      "description": "OK",
      "gameID": 6000,
      "serialGameIdx": 1,
      "cardNumber": 20,
      "givenCardsNumber": 0,
      "hour": 10
    }
  ]
}
```

PreorderedReport

This method returns the list of cards preordered by player.

Request parameters

Property	Type	Description	Status
secureLogin	string	Login to access the API	Required
playerID	string	Unique player identifier.	Required
dateFrom	Number	Start date of the report. UNIX time in ms	Required
dateTo	Number	End date of the report. UNIX time in ms	Required
language	string	Player's language	Optional
roomID	Number	ID of the room	Optional

Example request

```
https://{bingo API domain}/BingoIntegration/BingoGameAPI/PreorderedReport
{
  "secureLogin": "securelogin",
  "playerID": "16243a2ec4218f2",
  "dateFrom": 1544433758165,
  "dateTo": 1544606558165,
  "language": "no",
  "roomID": 1442
}
```

Response parameters

Property	Type	Description	Status
error	number	Response status.	Required
description	string	Response status short description.	Required
preorderedList	array of type 'PreorderedReport'		Required

Structure 'PreorderedReport'

Property	Type	Description	Status
transactionID	number	Reference of bet request in wallet API.	Required
sessionStartDate	number	Date and time of the draw. UNIX time in ms	Required
gameID	number	ID of the game. (Bingo game ID)	Required
gameName	string	Name of the game.	Required
bingoType	integer	Bingo game type (75 or 90)	Required
roomID	number	ID of the room.	Required
roomName	string	Name of the room.	Required
cardNumbers	integer	Number of the cards.	Required
givenCardsNumber	Integer	The number of cards given to player in addition (relevant for Buy X get Y games).	Required

cardCost	double	Card price.	Required
purchaseMode	string	C – Card / Ticket	Required
		F – Fixed price	
		S – Strips Fixed price	
		T – Strips Ticket price	
roundID	number	Round ID for player transaction in wallet API.	Required
bingoRoundID	number	Round Identifier for bingo round from Provider side where player will play preordered cards.	Required
currency	string	Currency of the player (ISO 4217)	Required
prize	double	Amount in player currency of prize for a round.	Required
jackpot	double	Amount in player currency of jackpot for a round.	Required

Example 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 } ] }
```


CancelPreorderedCards

This method cancels the cards preorder made by player.

Request parameters

Property	Type	Description	Status
secureLogin	String	Login to access the API	Required
playerID	String	Unique player identifier.	Required
cancelPreorderedList	array of type 'CancelPreordered'	List of cards to cancel the pre-order.	Required

Example request

https://{bingo API domain}/BingoIntegration/BingoGameAPI/CancelPreorderedCards

```
{
  "cancelPreorderedList": [
    {
      "transactionID": 10001000
    }
  ],
  "playerID": "16243a2ec4218f2",
  "secureLogin": "securelogin"
}
```

Structure 'CancelPreordered'

Property	Type	Description	Status
error	string	Response status for each canceled transaction.	Required
description	string	Response status short description.	Required
transactionID	number	Reference of bet request in wallet API.	Required

Response parameters

Property	Type	Description	Status
error	string	Response status for request.	Required
description	string	Response status short description.	Required
cancelPreorderedList	array of type 'CancelPreordered'	The resulting list of pre-order cancellation.	Required

Example response

```
{
  "cancelPreorderedList": [
    {
      "description": "OK",
      "error": "0",
      "transactionID": 20001661407
    }
  ],
  "description": "OK",
  "error": 0
}
```

JackpotWinners

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

Request parameters

Property	Type	Description	Staus
secureLogin	String	Login to access the API	Required
date	Number	Day for requesting bingo jackpot winners. UNIX time in ms.	Required
language	string	Player’s language	Optional

Example request

```
https://{bingo API domain}/BingoIntegration/BingoGameAPI/JackpotWinners
{
  "date": 1526992641791,
  "language": "en",
  "secureLogin": "login1526992641791"
}
```

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 or 90)	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

```
{"jackpotWinners":[{"playerID":"playerID","nickname":"player566322","amount":110.5,"dateTime":0,"jackpotName":"Jackpot 2","roomID":200,"bingoType":75,"gameName":"Game 122","roundID":50175,"currency":"USD"}],"error":0,"description":"OK"}
```

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

```
https://{bingo API domain}/BingoIntegration/BingoGameAPI/ChangeNickname
{
  "playerID": "playerID",
  "nickname": "NewNickname",
  "secureLogin": "login1526992641791"
}
```

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:

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

{
  "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 error.	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 wwq sky1 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:

```
https://{bingo API domain}/BingoIntegration/BingoGameAPI/BingoRoundHistory
{
  "roundID": 1000,
  "language": "en",
  "secureLogin": "login1526992641791"
}
```

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 cards from pack.	Required
currency	string	Currency of the player (ISO 4217). Examples: EUR' or USD'.	Optional

Example response:

```
{"error":0,"description":"OK","gameName":"vcb wwg 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","cu
rrency":"USD","prizeAmount":1.67,"jpAmount":167,"cardID":79000},{ "playerID":"16243a2ec4218f2","nickname"
:"sk_en_65","currency":"USD","prizeAmount":1.67,"jpAmount":167,"cardID":79006},{ "playerID":"16243a2ec421
8f2","nickname":"sk_en_65","currency":"USD","prizeAmount":1.67,"jpAmount":167,"cardID":79005}]}], "bought
Cards":[{"playerID":"16243a2ec4218f2","boughtCardsNumber":10,"givenCardsNumber":10,"freeCardsNumber":10,
"cardCost":1,"currency":"USD"}]}
```

Error Codes

Table below contains the error codes that should be returned by PROVIDER in the response.

Code	Description
0	Success.
1	Insufficient balance. The error should be returned in the response on the Bet request.
5	Invalid hash code. Should be returned in the response on any request sent by OPERATOR if Incorrect secureLogin or/and securePassword.
7	Empty mandatory field. Should be returned in the response if required field is empty.
8	Invalid parameter. Should be returned in the response if field is used with incorrect name, data type or value.
9	Duplicate nickname. Should be returned in the response on change nickname if same nickname already is used in system for current casino.
10	Player not found. Should be returned if player is not found at provider side
11	Currency code is incorrect or unsupported. Should be returned if currency in request is unsupported or has incorrect code.
100	Internal server error. OPERATOR will return this error code if their system has internal problem and cannot process the request.
101	Preorder cards limit exceeded. Should be returned when total number of cards for needed bingo game is more than limit, set for this game.
102	Preorder of cards is not allowed. Pre-order of cards is not allowed at this point of time for the selected game.
103	Preorder game is not available. Returned if game is not available anymore (Due to scheduling etc)
105	The requested data not found. Returned when any data could be found by received request (for example incorrect timestamp and/or bingo roundid in request was received)
106	Cancel preorder cards not found. The cards were not found according to the criteria.
107	Incorrect number of cards. Should be returned when number of cards for fixed price game is less than value which was set for this game
109	Wallet transaction failed. Should be returned when BET transaction to operator side fails on some reason

16.5 Bingo Free rounds API

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.

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://{bingo API service domain}/BingoIntegration/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

Needed methods and specification are described in chapters [5.1–5.6](#) in section [V. Simple Free Rounds API](#) of current document