

Fundist API v285(wr)

Fundist

Exported on 11/28/2025

Table of Contents

Contents	5
Change History.....	6
1. Introduction.....	15
2. System definitions	16
3. Quick Start Guide	17
Mandatory steps to launch a game:	17
3.1 Game Catalogue	17
3.2 Player Account.....	17
3.3 Game Launch	18
3.4 User's balance.....	18
3.4.1 Wallet types.....	18
3.4.2 One Wallet Holistic Picture	19
3.4.3 Separate Wallet Holistic Picture	20
API Request Specification.....	22
4. New user creation and/or update	23
4.1 User information update	24
4.2 User's phone verification.....	25
5. Users management extra functionality	26
5.1 Enable of user	26
5.2 Disable of user	26
5.3 Get user data.....	27
5.4 Get list of user categories	28
5.5 Set / Unset category to user	29
5.6 Set tags	30
5.7 Unset tags	30
5.8. User's nick update	31
6. Game Lists	33

6.1. Get game categories	33
6.2. Get available games	34
6.3 Get all game catalog related information	36
6.4 Get tables lobby state.....	38
6.5 Get sorted games	40
6.6 Get sort list of games by sort type	41
7. Autolobby	43
7.1 Access in Fundist	43
7.2 How to launch the application of autolobby	44
7.3 Minimal Lobby Launch Template	45
7.4 Repainting	46
8. Games Launch	48
8.1 Check for player country	48
8.2 Authorization with HTML fragment.....	48
8.3 Kill authorization (end active sessions)	50
9. Statistics.....	51
9.1 Universal Bets Summary	51
9.2 Stats Bets	52
9.3 Stats Detailed.....	54
9.4 Game Detailed.....	55
9.5 Loyalty payments.....	57
9.6 Bonus select.....	58
9.7 Bonus cancel.....	59
9.8 Subscribe tournament.....	59
9.9 Order store item	60
10. Error handling.....	62
10.1 Retrieve response of previously executed request	62
10.2 Retrieve information of previously executed request	62
11. Providers additional info	64

11.1 Get live providers table info	64
11.2 Get Amusnet / EGT Digital jackpots widget*	65
11.3 Get jackpots list	67
11.4 Get Min/Max bet (PGSoft only)	68
11.5 Get Sport bets	71
12. Frequently asked questions (FAQ)	75
13. Appendices	76
Appendix 1. Error codes	77
Appendix 2. Currency codes	82
Appendix 3. Language codes	86

Contents

Change History

2. 0	13 Feb 2013	Documentation is split into several documents: <ul style="list-style-type: none"> • Generic API • System-specific parameters
2. 0 1	26 Mar 2013	Extended language list
2. 0 2	01 May 2013	Extended user authentication with “IsMobile” parameter support Added color coding
2. 0 3	29 Aug 2013	Added clarification of [TID]
2. 0 4	11 Sep 2013	Added optional Timezone parameter for User/Add and User/AuthHTML Added previously not documented Nick parameter to User/Add
2. 0 5	24 Sep 2013	Extended User/Add with number of new parameters
2. 0 6	25 Sep 2013	Added “Funds Transfer Safety” section
2. 0 7	09 Nov 2013	Dropped deprecated User/Auth request
2. 0 8	13 Nov 2013	Clarified format of Amount parameter Added previously not documented RegistrationIP parameter for User/Add Updated error codes Misc. formatting

2. 0 9	26 Jan 2014	Minor correction of User/AuthHTML Hash calculation due to issue introduced in Rev 2.07
2. 1 0	01 Apr 2014	Added optional ExtParam paramter to User/AuthHTML
2. 1 1	04 May 2014	Corrected links to include more proper path starting from /System/ Added previously not documented UserIP parameter for User/AuthHTML Added User/KillAuth request. Corrected references to Appendix and product-specific Addendum Added Holistic Picture
2. 1 2	07 Jul 2014	Added User/AuthHTML API call
2. 1 3	11 Aug 2014	Added support for Demo play without user creation in User/*Auth* requests for supported systems
2. 1 4	11 Aug 2014	Finally, internal dynamic game list interface got exposed as Game/* family of public API
2. 1 5	23 Sep 2014	Changed format of Games/List: "HasMobile" is replaced with "MobilePageCode" field
2. 1 6	16 Oct 2014	Added "AffiliateID" support to Stats/BetsSummary Added previously not documented "User/Update" request
2. 1 7	06 Nov 2014	New error codes related to payment system integration Lao language added
2. 1 8	18 Dec 2014	Added ExtParam support to User/KillAuth

2. 1 9	20 Jan 2015	Changed optional parameters to become mandatory: User/Add: Language, Gender, Country, DateOfBirth, RegistrationIP User/*Auth*: Page, UserIP
2. 2 0	20 May 2015	Added ExternalCode and MobileExternalCode to Game/List
2. 2 1	14 Jul 2015	Added games jackpots API request
2. 2 2	20 Aug 2015	100 error code added
2. 2 3	04 Nov 2015	666 code change to 600 for non financial errors
2. 2 4	25 Nov 2015	Ukrainian language is added to the list of supported languages. Minor erratum corrections
2. 2 5	12 Jan 2016	Game list image caching note is added
2. 2 6	18 Feb 2016	Added loyalty support for Balance/Set, Balance/Get, Balance/Withdraw API requests
2. 2 7	21 Sep 2016	Added a note regarding parameter value URL encoding
2. 2 8	09 Jan 2017	Added User/Enable and User/Disable API call

2. 2 9	26 Jan 2017	Minus sign added to allowed char list for account login
2. 3 0	20 Feb 2017	Mandatory params list cleanup for User/Add request
2. 3 1	04 May 2017	Section rearrangement
2. 3 2	19 May 2017	Added “Hour” support to Stats/BetsSummary
2. 3 3	1 June 2017	Description added for UserAutoCreate option on game launch
2. 3 4	23 Aug 2017	Added description for Stats/Bets, Stats/Detailed API calls
2. 3 5	29 Aug 2017	Description regarding country check on game launch is added. Note regarding country auto detection on account creation is added. Minor erratum corrections
2. 3 6	4 Sep 2017	Minor erratum corrections
2. 3 7	18 Sep 2017	Section rearrangement
2. 3 8	13 Nov 2017	Added description for Stats/GameDetails API call
2. 3 9	22 Nov 2017	Minor erratum corrections

2. 4 0	29 Nov 2017	Section rearrangement. Minor erratum corrections
2. 4 1	12 Dec 2017	Example for query string added. Supported HTTP method for API calls are listed
2. 4 1	14 Dec 2017	Additional description added for [CASINO_SERVER_IP] parameter
2. 4 3	18 Dec 2017	Supported currency list is updated. FAQ section is introduced
2. 4 4	16 Jan 2018	A note regarding password for Demo mode is added. And also a note, how to identify wherever game has demo mode available
2. 4 5	16 Okt 2018	Added Providers/TablesInfo calls for EVO and AuthenticGaming
2. 4 6	18 Jun 2019	Added description for Loyalty/Payment API call
2. 4 7	28 Jun 2019	Added description for Bonuses API calls
2. 4 8	12 Jul 2019	Added new parameters for Loyalty/Payment API call
2. 4 9	20 Aug 2019	Added Game/FullList API call description
2. 5 0	22 Aug 2019	Additional game launch code params, external game code and full image URL added for Game/FullList API call

2. 5 1	03 Oct 2019	Game/List API call description fulfilled with additional fields, that are present on API response
2. 5 2	29 Nov 2019	Added Tables/LobbyState API call description
2. 5 3	29 May 2020	New fields added to Game/List and Game/FullList response: "SubMerchantName", "SubSystem", "IDParent"
2. 5 4	02 July 2020	Description for Error code 35 was corrected and Error code 50 was added
2. 5 5	04 Aug 2020	Added WLCGames/Jackpots API call description
2. 5 6	06 Aug 2020	Added TableID field to response for Game/List and Game/FullList
2. 5 7	10 Sep 2020	More detailed description is introduced for RegistrationIP/UserIP params, as well as [IP] is replaced with [CASINO_SERVER_IP] to avoid confusion, which definition is for server IP and which is for player IP
2. 5 8	07 Jan 2021	Description for Referer param has been added for User/AuthHTML API call
2. 5 9	15 Jan 2021	Description for Game/Sorting has been added
2. 6 0	17 Feb 2021	Description for UniversalLaunch param has been added for User/AuthHTML API call
2. 6 1	01 Mar 2021	Description for Providers/UpdateNick has been added

2. 6 2	22 Mar 2021	“Hour” support to Stats/BetsSummary has been removed. A note regarding DNS TTL is added for 2. System Definitions.
2. 6 3	14 May 2021	Hour Description of CustomSort and RPT parameters were added to Game/List and Game/FullList Description of Weight parameter was added to Game/FullList Added additional parameters for loyalty balance amount in GameDetails
2. 6 4	02 Jun 2021	Wallet Type field with Holistic Picture for each of wallet types was added to Introduction Funds Transfer Safety section moved to Balances and Transfers
2. 6 5	20 Aug 2021	Danish language code has been corrected
2. 6 6	02 Jun 2022	Categories and tags sections are added to 2. Enable/Disable of user paragraph.
2. 6 7	01 Jul 2022	Added Get user data section to 2. Enable/Disable of user paragraph.
2. 6 8	10 Oct 2022	Autolobby configuration description is added.
2. 6 9	24 Oct 2022	Description for Tournament/Select and Store/Buy has been added
2. 7 0	10 Jan 2022	New field IDMerchantsCurrencies is introduced as a part of game data in a response for Game/FullList API call
2. 7 1	13 Mar 2023	User/PhoneVerify API call description has been added
2. 7 2	20 Mar 2023	Game/Sorts API call description has been added

2. 7 3	10 Apr 2023	Optional param PaySystem description for Balance/Set request has been added
2. 7 4	22 May 2023	MinBetDefault, MaxBetDefault, MaxMultiplier fields are added to json response of Game/List and Game/FullList API calls
2. 7 5	29 June 2023	Amusnet Interactive jackpots widget request description added
2. 7 6	18 Sept 2023	Amusnet Interactive jackpots widget request description updated
2. 7 7	24 Oct 2023	New field Freeround is introduced as a part of game data in a response for Game/FullList API call
2. 7 8	13 Dec 2023	Get Min/Max bet (PGSoft only) is added
2. 7 9	2 Jan 2024	Field list for API method Game/Sorting is updated
2. 8 0	10 Oct 2024	Get sport bets history added
2. 8 1	20 Nov 2024	Parameter Currencies is added to Providers/TablesInfo
2. 8 2	18 March 2025	Parameter WithTechnical is added to Game/Categories request
2. 8 3	06 Aug 2025	EGT Digital jackpots widget request description added

2. 8 4	10 Nov 2025	Information about error code 51 added
2. 8 5	26 Nov 2025	11.5 Get Sport bets - now is also available for Digitain. Also, error codes list in Appendix 1 is updated

1. Introduction

This document describes API of FUNDIST.ORG system.

For access setup and technical support please contact FUNDIST.ORG.

This document is intended only for authorized partners of FUNDIST.ORG, and it's forbidden to distribute the document without written permission of FUNDIST.ORG representatives.

Testing environment is available for API integration development and verification.

Before entering production, please make sure your system has successfully passed all integration tests, especially for error handling. Please contact FUNDIST.ORG technical support to get the latest integration test script.

API setup on FUNDIST.ORG side requires full list of IPs for white-listing.

2. System definitions

[SERVER] – API server domain.

[KEY] – API access key.

[PWD] – API access password.

[CASINO_SERVER_IP] – API client IP or '0.0.0.0'. API client here means server, from which API calls are made. Basically [CASINO_SERVER_IP] – is server IP address, from which actual API call is made. Alternatively you could use '0.0.0.0' in case, when real external server IP is not known at Hash parameter calculation time. This value is used ONLY for Hash parameter calculation.

[HASH] – hash value for current request, which basically is an API call signature.

[TID] – Unique transaction ID (for whole history of requests, 32 chars max)

[URL] – URL for API request.

[SYSTEM] – System ID of target Merchant.

{Param} – optional parameter (without brackets)

To make API call, client calls a specially defined [URL]. All requests are case-sensitive and require the exact format, described in Chapter API Request Specification. Every request has a defined response. Usually, response consists of response codes and additional information, separated by commas. Response codes are defined in Appendix 1.

Important notes:

Special characters (like space character, "+", "=", etc) in parameter values should be properly URL encoded according to RFC2396.

Client's system must honour the time period during which resource records are cached - TTL for used server name.

Query string note: API parameters start from "?&", but not traditional "?". For example: [https://\[SERVER\]/System/Api/\[KEY\]/Example/Call/?&TID=\[TID\]&Hash=\[HASH\]&...](https://[SERVER]/System/Api/[KEY]/Example/Call/?&TID=[TID]&Hash=[HASH]&...)

Supported HTTP methods for API calls: GET, POST.

3. Quick Start Guide

All API requests (AuthHTML, Game/List and others from our Fundist API doc) need to be passed to our server (apitest.fundist.org¹)

Once you start playing the game our backend will send One Wallet requests to your callback URL (endpoint which points on your server and where we send debit, credit, ping and balance requests). Callback should be configured on your side and provided to us.

In case of one (=single/seamless) wallet integration, balance is managed on your side.

Mandatory steps to launch a game:

3.1 Game Catalogue

We offer you a game catalogue, available over API. To get all possible information related to games you can use Game/FullList API call (3.3. Get all game catalog related information) – game categories, game list, restricted country list, enabled provider list; currency list, supported by provider.

An example of such request on test: <https://apitest.fundist.org/System/Api/YourAPIKEY/Game/FullList?&TID=YourTID&Hash=YourHASH>.

Hash calculation can be found in our API doc. Each API request has its own HASH calculation formula.

API Key value is provided to you in our email, as well as your API Password.

TID needs to be created on your side.

3.2 Player Account

In order to run a game, you need a player account on our side.

The easiest way to accomplish that – to make API call for game launch (User/AuthHTML) with additional params:

UserAutoCreate=1&Currency=<Currency ISO Code>&Country=<ISO3Code>

Important note: Player currency cannot be changed.

¹ <http://apitest.fundist.org/>

3.3 Game Launch

User/AuthHTML (5.3. Authorization with HTML fragment) returns HTML/JS fragment, that should be embedded into site page

Country check is made on this step, by using geoIP, based on player IP value, provided by casino.

As an example – you need to launch a certain game from a certain provider. For this you need to perform a Game/FullList request and find a game you would like to launch.

For example, you would like to launch Crazy Time game from Evolution provider. After performing Game/FullList request you should check "ProviderID" and "ID" value in "merchants" part (for Evolution it is 998) and "PageCode" value which is crazytime:CrazyTime0000001.

Using this info your API request to launch the game Crazy Time from Evolution would be:

[https://apitest.fundist.org/System/Api/\[KEY\]/User/AuthHTML/?&Login=\[LOGIN\]&Password=\[PASSWORD\]&System=998&TID=\[TID\]&Hash=\[HASH\]&Page=crazytime:CrazyTime0000001&UserIP=\[USERIP\]](https://apitest.fundist.org/System/Api/[KEY]/User/AuthHTML/?&Login=[LOGIN]&Password=[PASSWORD]&System=998&TID=[TID]&Hash=[HASH]&Page=crazytime:CrazyTime0000001&UserIP=[USERIP])

Login, Password, TID, HASH and UserIP are configured on your side.
API Key value is sent to you in our email.

By adding UserAutoCreate=1&Currency=<Currency ISO Code>&Country=<ISO3Code> to the AuthHTML request above you can automatically create a user. If a user is already created, this part can be skipped.

3.4 User's balance

API requests are not connected to user's balance – it is managed by wallet integration. It is described in our Fundist API Wallet docs.

You can try launching games with API requests first, then move to implementing wallet.

3.4.1 Wallet types

Basically there are two possible wallet types available: one wallet and separate wallet.

One Wallet may be also called single or seamless and is considered to be a **default type** for an API integration. In this case players balance is stored on the casino (client's) side.

Our system sends real-time debit, credit and balance requests during the game process through HTTPS protocol. Request body is sent as JSON object using POST method. All the game statistics is available to the client right away.

More details about one wallet integration are present in API One Wallet document. Please request our integration team to send the mentioned document if you haven't been provided with it yet.

Separate Wallet is another possible type when there are two independent wallets on both casino (client's) and our side.

In this scenario players' balance is stored on our side and is being managed by casino (client's) system via a set of balance requests. Game statistics should be retrieved additionally via API, or else via Reports section in Fundist Backoffice.

If you want to know more you can ask your Integration Manager to send you "Fundist API Separate Wallet" manual.

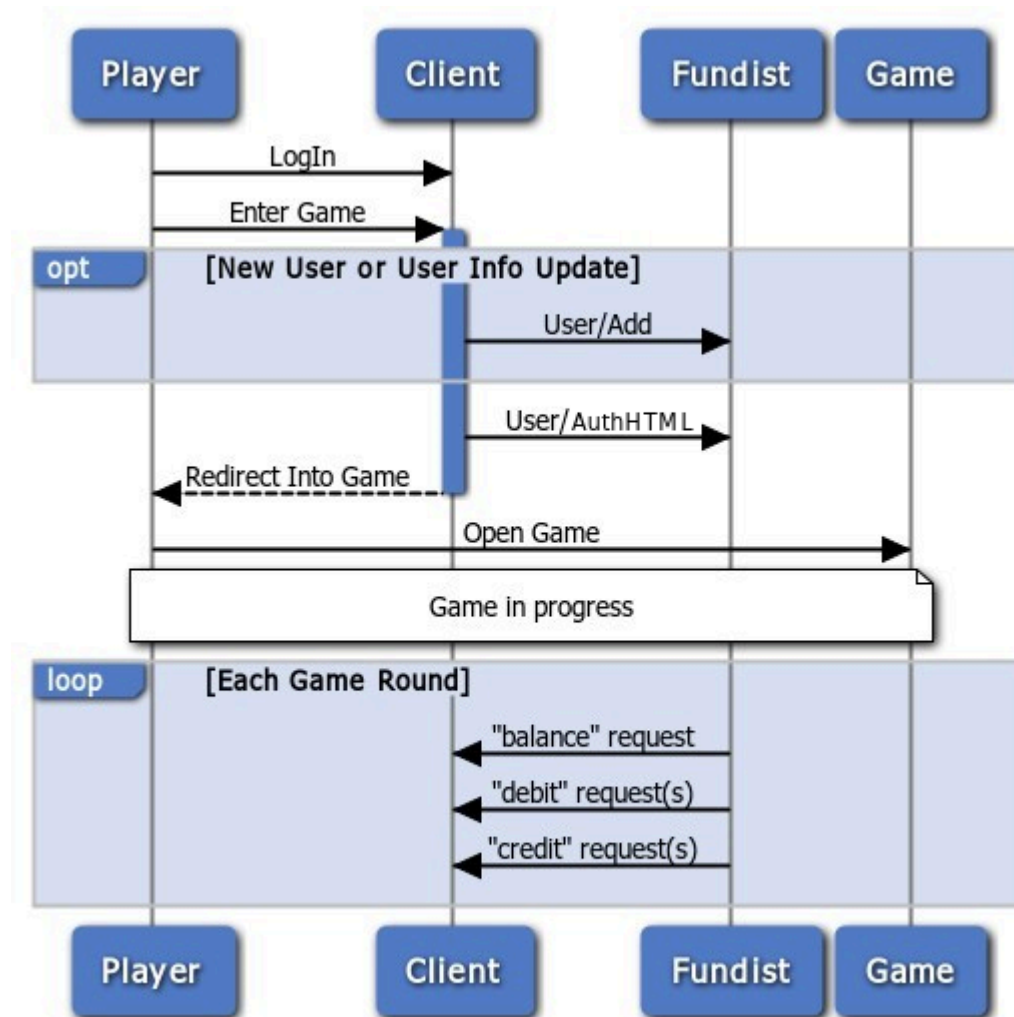
3.4.2 One Wallet Holistic Picture

Player – user's web browser

Client – client system with own website

Fundist – eGamings financial system, hidden from Player

Game – one of game product implementations, visible to Player



Callbacks are sent as JSON messages.

Four types of one wallet callbacks:

- ping
- balance,
- debit (usually bet),
- credit (usually win).

Cancellation might be of 2 types:

cancellation of duplicate request, that has been sent for example because of late response – with type debit/credit + i_rollback field that has tid (transaction ID, that should be rolled back).

We have a number of mandatory test scenarios that should be passed before going live. You can test your one wallet protocol implementation by using our OWClientTest JAR file provided to you in our email.

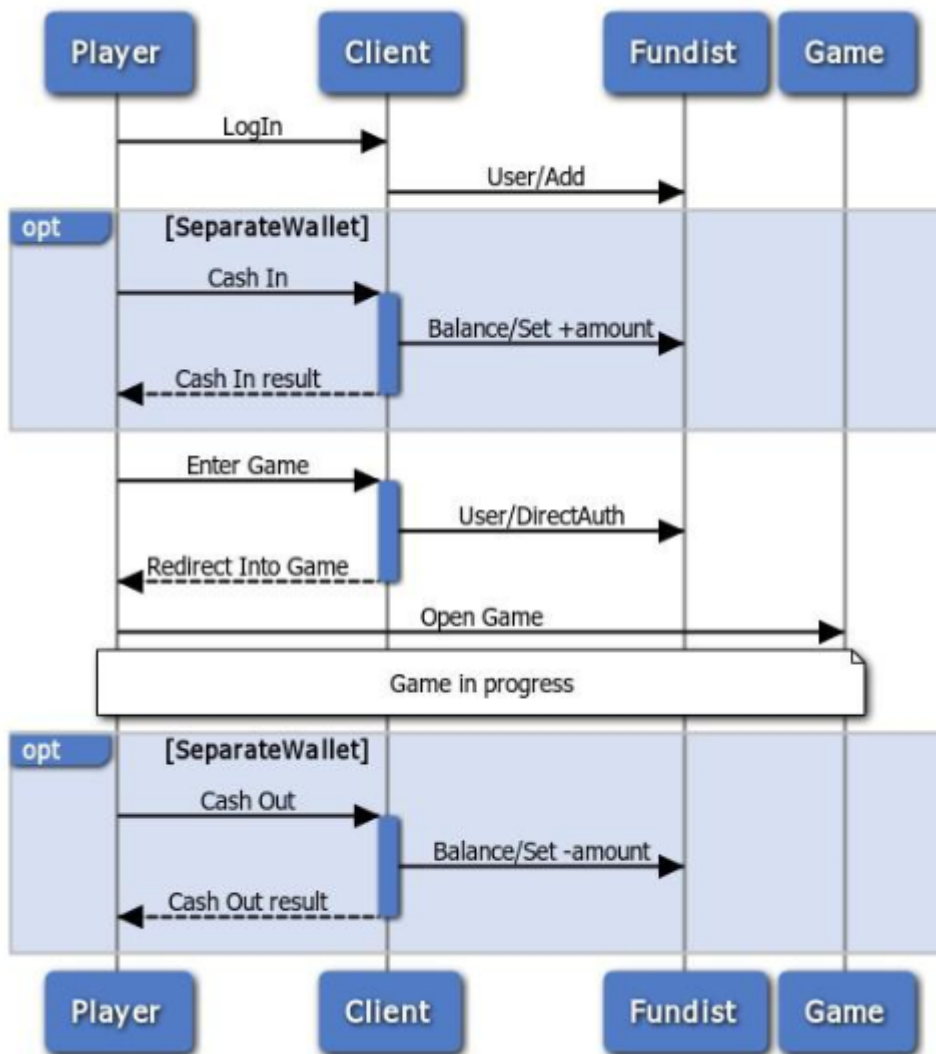
3.4.3 Separate Wallet Holistic Picture

Player – actual gambler's web browser

Client – client system with own website

Fundist – eGamings financial system, hidden from Player

Game – one of game product implementations, visible to Player



API Request Specification

4. New user creation and/or update

In order to launch a game, user account should be created first. Alternatively, user account could be created automatically by passing UserAutoCreate=1 for User/AuthHTML API calls.

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/User/Add/?

&Login=[LOGIN]

&Password=[PASSWORD]

&TID=[TID]

&Currency=[CURRENCY]

&Hash=[HASH]

&Language=[LANGUAGE]

&RegistrationIP=[REGIP]

{&Gender=[GENDER]}

{&Country=[COUNTRY]}

{&DateOfBirth=[DOB]}

{&Nick=[NICK]}

{&Timezone=[TIMEZONE]}

{&Name=[NAME]}

{&LastName=[LASTNAME]}

{&Phone=[PHONE]}

{&AlternativePhone=[ALTPHONE]}

{&City=[CITY]}

{&Address=[ADDRESS]}

{&Email=[EMAIL]}

{&AffiliateID=[AFFILIATEID]}

where:

[LOGIN] – User Login in API system. Allowed symbols: 0-9, a-z, A-Z and _-. Maximal length: 29 symbols. The login can consist only from digits, what allows using client's system unique user ID. In our back office, the user names are prefixed with special string NN_, where NN_ is always the same per API setup.

[PASSWORD] – User's password. It's an internal password shared only between FUNIDST.ORG and client's system. Normally, this password is a random generated string. Do NOT use the real password of user in client's system! Minimal length is 6. It should not contain the [LOGIN].

[CURRENCY] – The currency of user's account. It cannot be changed!

[LANGUAGE] – language code (Appendix 3). If present, user's language in database is updated.

[REGIP] – player registration IP address, which is valid IPv4 (e.g. 12.34.56.78) or valid IPv6 address (for example 2001:0db8:85a3:0000:0000:8a2e:0370:7334). This value is used to identify country by using GeoIP library when Country param is omitted from request params, to perform check against restricted country list at game launch.

[HASH] – MD5 checksum of the following data:

User/Add/[CASINO_SERVER_IP]/[TID]/[KEY]/[LOGIN]/[PASSWORD]/[CURRENCY]/[PWD]

Optional parameters:

[NICK] – nick name, which prevents prompting on first entry in some merchants 10

[TIMEZONE] – timezone shift in minutes relative to UTC. Valid range is from -720 till +720.

[NAME] – first name

[LASTNAME] – last name

[GENDER] – male or female

[PHONE] – primary phone

[ALTPHONE] – alternative phone

[COUNTRY] – ISO 2 or 3 symbol code.

NOTE: In case Country param is not passed or passed with unknown country code for Fundist.ORG system on User/Add API call, country will be auto detected by using geoip library from RegistrationIP value.

[CITY] – city name

[ADDRESS] – home address

[EMAIL] – e-mail. Must not be changed or user may be rejected by some merchants until restored

[DOB] – YYYY-MM-DD

[AFFILIATEID] – third-party affiliate ID

Server response:

1. On success: 1
2. On error, and error code and description is returned, according to Appendix 1
3. On duplicate login, database is updated with new information and success status is returned

4.1 User information update

User/Update – is an alias for User/Add. Please use this request for semantically more correct naming.

Note:

[HASH] – MD5 checksum of the following data:

User/Update/[CASINO_SERVER_IP]/[TID]/[KEY]/[LOGIN]/[PASSWORD]/[CURRENCY]/[PWD]

4.2 User's phone verification

This API call is used for user's phone number verification.

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/User/PhoneVerify/?

&Login=[LOGIN]

&Password=[PASSWORD]

&Phone = [PHONE]

&TID=[TID]

&Hash=[HASH]

where:

[LOGIN] – User Login in API system.

[PASSWORD] – User password in API system.

[PHONE] – User phone in API system.

[HASH] – MD5 checksum of the following data:

User/PhoneVerify/[IP]/[TID]/[KEY]/[LOGIN]/[PASSWORD]/[PWD]

Server response:

1. On success: 1,Phone verified
2. On error, and error code and description is returned, according to Appendix 1

5. Users management extra functionality

Note: this chapter is not mandatory, but it can be helpful for you.

5.1 Enable of user

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/User/Enable/?`

`&Login=[LOGIN]`

`&TID=[TID]`

`&Hash=[HASH]`

`{&Reason=[REASON]}`

where:

[LOGIN] – User Login in API system. Allowed symbols: 0-9, a-z, A-Z and _-. Maximal length: 29 symbols. The login can consist only from digits, what allows using client's system unique user ID. In our back office, the user names are prefixed with special string NN_, where NN_ is always the same per API setup.

[HASH] – MD5 checksum of the following data:

`User/Enable/[CASINO_SERVER_IP]/[TID]/[KEY]/[LOGIN]/[PWD]`

Optional parameters:

[REASON] – reason for enabling of user

Server response:

1. On success: 1,Active
2. On error, and error code and description is returned, according to Appendix 1

5.2 Disable of user

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/User/Disable/?`

`&Login=[LOGIN]`

`&TID=[TID]`

`&Hash=[HASH]`

`{&Reason=[REASON]}`

where:

[LOGIN] – User Login in API system. Allowed symbols: 0-9, a-z, A-Z and _-. Maximal length: 29 symbols. The login can consist only from digits, what allows using client's system unique user ID. In our back office, the user names are prefixed with special string NN_, where NN_ is always the same per API setup.

[HASH] – MD5 checksum of the following data:

User/Disable/[CASINO_SERVER_IP]/[TID]/[KEY]/[LOGIN]/[PWD]

Optional parameters:

[REASON] – reason for disabling of user

Server response:

1. On success: 1,Inactive
2. On error, and error code and description is returned, according to Appendix 1

5.3 Get user data

Allow to get user data.

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/User/GetUserData/?

&Login=[LOGIN]

&TID=[TID]

&Hash=[HASH]

where:

[LOGIN] – User Login in API system. Allowed symbols: 0-9, a-z, A-Z and _-. Maximal length: 29 symbols. The login can consist only from digits, what allows using client's system unique user ID. In our back office, the user names are prefixed with special string NN_, where NN_ is always the same per API setup.

[HASH] – MD5 checksum of the following data:

User/GetUserData/[CASINO_SERVER_IP]/[TID]/[KEY]/[LOGIN]/[PWD]

Server response:

```
1. On success: 1,
{
  "UserId": "123",
  "Nick": "",
  "ExtLogin": "",
  "Timezone": "0",
  "Name": "Mik",
  "LastName": "Ch",
  "MiddleName": "",
```

```

"Gender": "male",
"Phone": "-",
"AlternativePhone": "-",
"City": "",
"Address": "",
"Email": "test@test.com2",
"DateOfBirth": "1980-01-01",
"IDNumber": "",
"IDIssueDate": "2020-01-01",
"IDIssuer": "",
"PlaceOfBirth": "",
"CityOfRegistration": "",
"AddressOfRegistration": "",
"IndexOfRegistration": "",
"AffiliateID": "",
"RegistrationIP": "",
"AffiliateSystem": "",
"AffiliateClickID": "",
"GiftBundleID": null,
"PostalCode": "123654",
"Pincode": "",
"Iban": "",
"BankName": "",
"Swift": "",
"BankAddress": "",
"BranchCode": "",
"PhoneVerified": "0",
"BonusCode": null,
"BonusID": null,
"EmailAgree": "0",
"SmsAgree": "0",
"Comment": "",
"ValidationLevel": "not-secure"
}

```

2. On error, and error code and description is returned, according to Appendix 1

5.4 Get list of user categories

Returns a list of categories that can be assigned to the user. Categories have the same function as tags. Only one category for one user can be set.

[URL] has a form of:

```

https://[SERVER]/System/Api/[KEY]/Categories/List/?
&TID=[TID]
&Hash=[HASH]

```

² <http://test.com/>

where:

[HASH] – MD5 checksum of the following data:

Categories/List/[CASINO_SERVER_IP]/[TID]/[KEY]/[PWD]

Server response:

1. On success: 1, *{json string of categories array as ID => Name}*
2. On error, and error code and description is returned, according to Appendix 1

Example of error response:

0, *Empty list*

5.5 Set / Unset category to user

Assigns a category to the user or removes it.

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/User/SetCategory/?

&TID=[TID]

&Login=[LOGIN]

&IDCategory=[IDCATEGORY]

&Hash=[HASH]

where:

[LOGIN] – User Login in API system. Allowed symbols: 0-9, a-z, A-Z and _-. Maximal length: 29 symbols. The login can consist only from digits, what allows using client's system unique user ID. In our back office, the user names are prefixed with special string NN_, where NN_ is always the same per API setup.

[IDCATEGORY] – ID of the category to be assigned. 0 in case of removal current category

[HASH] – MD5 checksum of the following data:

User/SetCategory/[CASINO_SERVER_IP]/[TID]/[KEY]/[LOGIN]/[PWD]

Server response:

1. On success: 1, *Ok*
2. On error, and error code and description is returned, according to Appendix 1

Example of error response:

14, *Wrong incoming params (Category not found)*

14, *Wrong incoming params (Category already set)*

5.6 Set tags

Assigns a tags (one or more) to the user. IDs of tags = IDs of categories

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/User/SetTags/?`

`&TID=[TID]`

`&Login=[LOGIN]`

`&IDTags=[IDTAGS]`

`&Hash=[HASH]`

where:

[LOGIN] – User Login in API system. Allowed symbols: 0-9, a-z, A-Z and _-. Maximal length: 29 symbols. The login can consist only from digits, what allows using client's system unique user ID. In our back office, the user names are prefixed with special string NN_, where NN_ is always the same per API setup.

[IDTAGS] – array of tag IDs to be assigned

[HASH] – MD5 checksum of the following data:

`User/SetTags/[CASINO_SERVER_IP]/[TID]/[KEY]/[LOGIN]/[PWD]`

Server response:

1. On success: 1, *Ok*
2. On error, and error code and description is returned, according to Appendix 1

Example of error response:

14,Wrong incoming params (Tags not found)

14,Wrong incoming params (Tags already set)

5.7 Unset tags

Removes tags (one or more) from the user.

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/User/UnsetTags/?`

`&TID=[TID]`

`&Login=[LOGIN]`

`&IDTags=[IDTAGS]`

&Hash=[HASH]

where:

[LOGIN] – User Login in API system. Allowed symbols: 0-9, a-z, A-Z and _-. Maximal length: 29 symbols. The login can consist only from digits, what allows using client's system unique user ID. In our back office, the user names are prefixed with special string NN_, where NN_ is always the same per API setup.

[IDTAGS] – array of tag IDs to be removed

[HASH] – MD5 checksum of the following data:

User/UnsetTags/[CASINO_SERVER_IP]/[TID]/[KEY]/[LOGIN]/[PWD]

Server response:

1. On success: 1, *Ok*
2. On error, and error code and description is returned, according to Appendix 1

Example of error response:

14,Wrong incoming params (Tags not found)

14,Wrong incoming params (Tags already set)

5.8. User's nick update

This API call is intended to change users nickname within the gameplay. Please note that this request can be used only for Evolution games.

[URL] has a form of:

[https://\[SERVER\]/System/Api/\[KEY\]/Providers/UpdateNick? \(see page 5\)&Login=\[LOGIN\]](https://[SERVER]/System/Api/[KEY]/Providers/UpdateNick? (see page 5)&Login=[LOGIN])

&Password=[PASSWORD]

&Provider=[PROVIDER]

&Nick=[NICK]

&TID=[TID]

&Hash=[HASH]

where:

[HASH] – MD5 checksum of the following data:

Providers/UpdateNick/[CASINO_SERVER_IP]/[TID]/[KEY]/[Login]/[Password]/[Nick]/[PWD]

[PROVIDER] – EvoSW

Server response:

1. On success: 1
2. On error, and error code and description is returned, according to Appendix 1

6. Game Lists

FUNDIST.ORG provides unique dynamic game lists, ready to be used directly in web sites. Of course, sane result caching for performance and load reasons is strongly recommended. We recommended to update the game list on daily basis.

The major advantage over all competitors is that categories and games are presorted in recommended order. However, each client can override the default sorting order, using our FUNDIST.ORG backoffice operator account.

The game list is continuously maintained by our staff.

6.1. Get game categories

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/Game/Categories/?`

`&TID=[TID]`

`&Hash=[HASH]`

`{&WithTechnical=[WITHTECHNICAL]}`

where:

[HASH] – MD5 checksum of the following data:

`Game/Categories/[CASINO_SERVER_IP]/[TID]/[KEY]/[PWD]`

[WITHTECHNICAL] - either 1 or 0. Optional parameter used to get technical categories of games. Game categories can be viewed in Fundist ->Games ->Categories section.

Server response:

1. On success, JSON formatted data is returned
2. On error, and error code and description is returned, according to Appendix 1

JSON response:

```
[
{
  "ID" : "globally unique numeric category ID",
  "Trans" : {
    "en" : "Name in English, always present",
    ...
  },
  "Tags" : [
    "main",
```

```
...
],
},
...
]
```

Note: "main" tag identifies primary category type. Non-primary categories are not necessary needed to be used/displayed.

6.2. Get available games

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/Game/List/?`

`&TID=[TID]`

`&Hash=[HASH]`

where:

[HASH] – MD5 checksum of the following data:

`Game/List/[CASINO_SERVER_IP]/[TID]/[KEY]/[PWD]`

Server response:

1. On success, JSON formatted data is returned
2. On error, and error code and description is returned, according to Appendix 1

JSON response:

```
[
{
  "AR": "Aspect ratio (width to height correlation) for a game in format W:H, for example 16:9",
  "Branded": "0/1 – flag that indicates, that game is branded",
  "Categories": [
    globally unique numeric category ID,
    ...
  ],
  "CustomSort": {
    "Lang": {"Custom game sort by language"}
  },
  "Description": ["ISO 2 character language code": "Game description on this language"],
  "ExternalCode": "External game code, that is sent over one wallet callback, if any; and also refers to game_desc field of Stats/*Auth*",
  "GSort": "Refers to global game sort order position on game catalog",
  "GSubSort": "Refers to custom assigned sort order position on game catalog for specific API",
  "HasDemo": "0/1 – flag, that indicates demo play support for particular game",
  "Freeround": "indicates whether a game has freerounds support or not, can be 1, 2, 3 or empty. 1-
```

```

desktop&mobile; 2- desktop; 3- mobile; empty - none",
"ID": "Refers to internal game ID",
"TableID": "Refers to internal table ID",
"IDCountryRestriction": "Refers to ID of restricted country list, under key `countriesRestrictions` ",
"ImageFullPath": "Game logo, full URL, see note below",
"ImageURL": "Game logo location at public servers",
"MerchantName": "Refers to gaming provider brand name",
"MinBetDefault": "default MinBet value, format example 1.00"
"MaxBetDefault": "default MaxBet value, format example 5.00"
"MaxMultiplier": "MaxMaltuplier value, format example 1.00"
"SubMerchantName": "If game is connected to sub merchant then refers to SubMerchant Name OR NULL",
"MobileAndroidPageCode": "Refers to Page parameter of User/*Auth* requests for game specific code for
Android, if any",
"MobilePageCode": "Refers to Page parameter of User/*Auth* requests",
"MobileExternalCode": "Refers to game_desc field of Stats/*",
"MobileWindowsPageCode": "Refers to Page parameter of User/*Auth* requests for game specific code for
Windows Phone, if any",
"PageCode": "Refers to Page parameter of User/*Auth* requests",
"SortPerCategory": {
    "Globally unique numeric category ID": "Refers to sort orderposition under specific game category",
    ...
},
"Status": "Refers to status of game record on game catalog, 1 – in case game is available",
"SuperBranded": "Flag, that refers to special group of games, that requires strict approval of all images used
for presenting and promotion of a game",
"System": "Refers to System (ie provider ID) parameter of User/*Auth* requests",
"SubSystem": "If game is connected to sub merchant then refers to SubMerchant ID OR NULL",
"RTP": "Game default RTP",
"Trans": {
    "en": "Name in English, always present",
    ...
},
...
]

```

Note: for performance reasons, we recommend to retrieve image (ImageURL) and cache on client web servers for players.

The direct linking is prohibited and we highly recommend to setup caching reverse proxy to the actual image URL.

Image path is relative (for example /agstatic/games/supergame.jpg), meaning, that full path would be <https://theverybestcasino.com/gstatic/games/supergame.jpg>, where theverybestcasino.com³ is casino actual domain. This requires additional configuration on web server, to setup caching reverse proxy under /agstatic/ path to the location [//agstatic.com/games/supergame.jpg](http://agstatic.com/games/supergame.jpg)⁴.

To send a second gamelist request, the first request must be completed. Otherwise, the second request will receive error 51 (check Appendix 1).

³ <http://theverybestcasino.com/>

⁴ <http://agstatic.com/games/supergame.jpg>

6.3 Get all game catalog related information

This API call is intended to get game categories, game list and restricted country list in one go.

[URL] has a form of:

```
https://[SERVER]/System/Api/[KEY]/Game/FullList/?
&TID=[TID]
&Hash=[HASH]
```

where:

[HASH] – MD5 checksum of the following data:

```
Game/FullList/[CASINO_SERVER_IP]/[TID]/[KEY]/[PWD]
```

Server response:

1. On success, JSON formatted data is returned. Please take into consideration, response body contains ONLY JSON, without "1," prefix.
2. On error, and error code and description is returned, according to Appendix 1

JSON response:

```
{
  "categories": [
    {
      "ID" : "globally unique numeric category ID",
      "Trans" : {
        "en" : "Name in English, always present",
        ...
      },
      "Tags" : [
        "main",
        ...
      ],
      "Name" : {
        "en" : "Name in English, always present",
        ...
      },
      "CustomSort" : {
        "Lang": {"Custom category sort by language"}
      },
    },
    ...
  ],
  "countriesRestrictions": [
    "0": {
      "Countries": [ "usa", "rus", .. , "ISO 3166 3 character country code" ],
      "ID": "Internal restricted country list ID",
      "IDMerchant": "Provider ID",
      "IsDefault": "1 – indicates, that restricted country list is default for all games for particular
provider",
      "Name": "Internal restricted country list name"
    },
    ...
  ]
}
```

```

    ...
  ],
  "games": [{
    "AR": "Aspect ratio (width to height correlation) for a game in format W:H, for example 16:9",
    "Branded": "0/1 – flag that indicates, that game is branded",
    "CategoryID": [
      "ID of game category, game belongs to",
      ...
    ],
    "CustomSort": {
      "Lang": {"Custom category sort by language"}
    },
    "Description": ["ISO 2 character language code": "Game description on this language"],
    "ExternalCode": "External game code, that is sent over one wallet callback, if any",
    "hasDemo": "0/1 – flag, that indicates demo play support for particular game",
    "Freeround": "indicates whether a game has freerounds support or not, can be 1, 2, 3 or empty. 1-
desktop&mobile; 2- desktop; 3- mobile; empty - none",
    "ID": "Refers to internal game ID",
    "TableID": "Refers to internal table ID",
    "IDCountryRestriction": "Refers to ID of restricted country list, under key `countriesRestrictions` ",
    "Image": "Game logo location, see note below",
    "ImageFullPath": "Game logo, full URL, see note below",
    "MerchantID": "Refers to System parameter of User/*Auth* requests",
    "SubMerchantID": "Refers to ID of sub-system if game is connected to merchant through one OR
`null` (mostly) or `0` if game is connected directly",
    "MobileAndroidPageCode": "Refers to Page parameter of User/*Auth* requests for game
specific code for Android, if any",
    "MobileAndroidUrl": "Refers to Page parameter of User/*Auth* requests for game specific code
for Android, if any; format: `<MerchantID>/<GameLaunchCode>` ",
    "MobileExternalCode": "External game code for mobile version, that is sent over one wallet
callback, if any",
    "MobilePageCode": "Refers to Page parameter of User/*Auth* requests",
    "MobileUrl": "Refers to Page parameter of User/*Auth* requests; format: `<MerchantID>/
<GameLaunchCode>`",
    "MobileWindowsPageCode": "Refers to Page parameter of User/*Auth* requests for game
specific code for Windows Phone, if any",
    "MobileWindowsUrl": "Refers to Page parameter of User/*Auth* requests for game specific code
for Windows Phone, if any; format: `<MerchantID>/<GameLaunchCode>`",
    "Name": {
      "en": "Name in English, always present",
      ...
    },
    "PageCode": "Refers to Page parameter of User/*Auth* requests",
    "Sort": "Order position on game catalog",
    "SortPerCategory": {
      "Category ID": "Order position on game category, default 0"
    },
    "SuperBranded": "Flag that indicates branded game, different set from `Branded`",
    "Url": "Refers to Page parameter of User/*Auth* requests for desktop version; format:
`<MerchantID>/<GameLaunchCode>`",
    "RTP": "Game default RTP",
    "MinBetDefault": "default MinBet value, format example 1.00"
  }

```

```

        "MaxBetDefault": "default MaxBet value, format example 5.00"
        "MaxMultiplier": "MaxMultiplier value, format example 1.00"
        "IDMerchantsCurrencies": "ID of a supported currency list or 0 in case of default one ",
    },
    ...
],
/* Provider list, available for API */
"merchants":{
    "ProviderID": {
        "Alias": "Provider name",
        "ID": "Provider ID, refers to System parameter of User/*Auth* requests if IDParent is NULL",

        "IDParent": "If provider is SUBprovider refers to ID of main Provider otherwise NULL",
        "Image": "Provider logo location, see note below",
        "Name": "Refers to internal provider name",
        "Weight": "Provider sorting weight",
    },
    ...
},
/* Supported currency list per provider */
"merchantsCurrencies":[{
    "Currencies": {"USD", "RUB", ..., "ISO 3 letter currency code"},
    "DefaultCurrency": "Refers to currency, that is set as auto conversion currency, in case desired
currency is not supported by provider",
    "ID": "Refers to internal supported currency list ID",
    "IDMerchant": "Provider ID, refers to System parameter of User/*Auth* requests",
    "IsDefault": "Flag, that indicates wherever this list is default for provider",
    "Name": "Refers to internal supported currency list name",
}],
]
}

```

Note: for performance reasons, we recommend to retrieve image (ImageURL) and cache on client web servers for players.

The direct linking is prohibited and we highly recommend to setup caching reverse proxy to the actual image URL.

Image path is relative (for example /agstatic/games/supergame.jpg), meaning, that full path would be <https://theverybestcasino.com/gstatic/games/supergame.jpg>, where theverybestcasino.com⁵ is casino actual domain. This requires additional configuration on web server, to setup caching reverse proxy under /agstatic/ path to the location [//agstatic.com/games/supergame.jpg](http://agstatic.com/games/supergame.jpg)⁶.

To send a second gamelist request, the first request must be completed. Otherwise, the second request will receive error 51 (check Appendix 1).

6.4 Get tables lobby state

Please note that this request can be used only for Evolution games.

⁵ <http://theverybestcasino.com/>

⁶ <http://agstatic.com/games/supergame.jpg>

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/Tables/LobbyState/?

&Tables=["table_id_1", "table_id_2", ...]

&TID=[TID]

&Hash=[HASH]

where:

["table_id_1", "table_id_2", ...] – list of required tables

[HASH] – MD5 checksum of the following data:

Tables/LobbyState/[CASINO_SERVER_IP]/[TID]/[KEY]/[PWD]

Server response:

1. On success, JSON formatted data is returned. Please take into consideration, response body contains ONLY JSON, without "1," prefix.
2. On error, and error code and description is returned, according to Appendix 1

JSON response:

```
{
  "< table_id_1 >": {
    "gameType ": "Game type",
    "name ": "Game name",
    "descriptions ": [],
    "open ": true,
    "betLimits ": {
      "EUR ": {
        "symbol ": "\u20ac",
        "min ": 1,
        "max ": 2000
      },
      ...
    },
    "operationHours ": {
      "type ": "FullTime"
    },
    "dealer ": {
      "dealerId ": "Dealer ID",
      "name ": "Dealer name"
    },
    "players ": 0,
    "videoSnapshot ": {
      "links ": { ... },
      "thumbnails ": { ... }
    },
    "display ": "on_mobile",
    "sitesAssigned ": [],
    "sitesBlocked ": []
  }
}
```

```

},
"< table_id_2 >": { ... }
}

```

6.5 Get sorted games

This API call is used to get the list of games marked as New or Popular. The list is dynamically formed depending on configuration made in Fundist: Games → Sorting, so the games in the list are displayed according to the sorting order.

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/Game/Sorting/?`

`&TID=[TID]`

`&Hash=[HASH]`

`&Type=new` or `Type=popular`

where:

[HASH] – MD5 checksum of the following data:

`Game/Sorting/[CASINO_SERVER_IP]/[TID]/[KEY]/[PWD]`

Server response:

1. On success, JSON formatted data is returned. Please take into consideration, response body contains ONLY JSON, without "1," prefix.
2. On error, and error code and description is returned, according to Appendix 1

JSON response:

```

[
  {
    "AR" : "Aspect ratio (width to height correlation) for a game in format W:H, for example 16:9",
    "Branded" : "0/1 – flag that indicates, that game is branded",
    "CategoryID" : [
      "ID of game category, game belongs to",
      ...
    ],
    "CustomSort" : {
      "Lang": {"Custom category sort by language"}
    },
    "Description" : ["ISO 2 character language code": "Game description on this language"],
    "ExternalCode": "External game code, that is sent over one wallet callback, if any",
    "hasDemo" : "0/1 – flag, that indicates demo play support for particular game",
    "Freeround": "indicates whether a game has freerounds support or not, can be 1, 2, 3 or empty. 1-
desktop&mobile; 2- desktop; 3- mobile; empty - none",
    "ID": "Refers to internal game ID",

```



```

    "TableID": "Refers to internal table ID",
    "IDCountryRestriction": "Refers to ID of restricted country list, under key `countriesRestrictions` ",
    "Image": "Game logo location, see note below",
    "ImageFullPath": "Game logo, full URL, see note below",
    "MerchantID": "Refers to System parameter of User/*Auth* requests",
    "SubMerchantID": "Refers to ID of sub-system if game is connected to merchant through one OR
'null' (mostly) or '0' if game is connected directly",
    "MobileAndroidPageCode": "Refers to Page parameter of User/*Auth* requests for game
specific code for Android, if any",
    "MobileAndroidUrl": "Refers to Page parameter of User/*Auth* requests for game specific code
for Android, if any; format: `<MerchantID>/<GameLaunchCode>` ",
    "MobileExternalCode": "External game code for mobile version, that is sent over one wallet
callback, if any",
    "MobilePageCode": "Refers to Page parameter of User/*Auth* requests",
    "MobileUrl": "Refers to Page parameter of User/*Auth* requests; format: `<MerchantID>/
<GameLaunchCode>`",
    "MobileWindowsPageCode": "Refers to Page parameter of User/*Auth* requests for game
specific code for Windows Phone, if any",
    "MobileWindowsUrl": "Refers to Page parameter of User/*Auth* requests for game specific code
for Windows Phone, if any; format: `<MerchantID>/<GameLaunchCode>`",
    "Name": {
        "en": "Name in English, always present",
        ...
    },
    "PageCode": "Refers to Page parameter of User/*Auth* requests",
    "Sort": "Order position on game catalog",
    "SortPerCategory": {
        "Category ID": "Order position on game category, default 0"
    },
    "SuperBranded": "Flag that indicates branded game, different set from `Branded`",
    "Url": "Refers to Page parameter of User/*Auth* requests for desktop version; format:
`<MerchantID>/<GameLaunchCode>`",
    "RTP": "Game default RTP",
    "MinBetDefault": "default MinBet value, format example 1.00"
    "MaxBetDefault": "default MaxBet value, format example 5.00"
    "MaxMultiplier": "MaxMultiplier value, format example 1.00"
    "IDMerchantsCurrencies": "ID of a supported currency list or 0 in case of default one "
},
...
]

```

6.6 Get sort list of games by sort type

The API request is only used for getting values of game sortings. Using it one receives an array as game ID => sorting value (with no additional logics of processing, only sorting data). The sorting value could be both a decimal and complex sorting object.

[URL] has a form of:

[https://\[SERVER\]/System/Api/\[KEY\]/Game/Sorts/](https://[SERVER]/System/Api/[KEY]/Game/Sorts/) (see page 5)?

SortType=[SORTTYPE]

&Type=[TYPE]

&TID=[TID]

&Hash=[HASH]

where:

[SORTTYPE] - sorting type. Possible values mentioned below. Mandatory field.

[TYPE] - restricts games by categories. Possible values: new or popular. Optional field.

[HASH] - MD5 checksum of the following data: Game/Sorts/[CASINO_SERVER_IP]/[TID]/[KEY]/[PWD]

Possible values of sorting types:

all - all sortings

auto - autosortings

global - global sortings

globalByCategories - global sortings by categories

globalByLanguages - global sortings by languages

globalByCountries - global sortings by countries

globalPerCategoriesByCountries - global sortings per categories by countries

local - local sortings

localByCategories - local sortings by categories

localByLanguages - local sortings by languages

localByCountries - local sortings by countries

localPerCategoriesByCountries - local sortings per categories by countries

Worth mentioning that SORTTYPE = auto - is auto sorting in section Merchant/Sorting, for it the parameter TYPE is mandatory. This is due to SORTTYPE = auto should return only autosortings which allowed only for categories new &popular from Merchant/Sorting.

Server response:

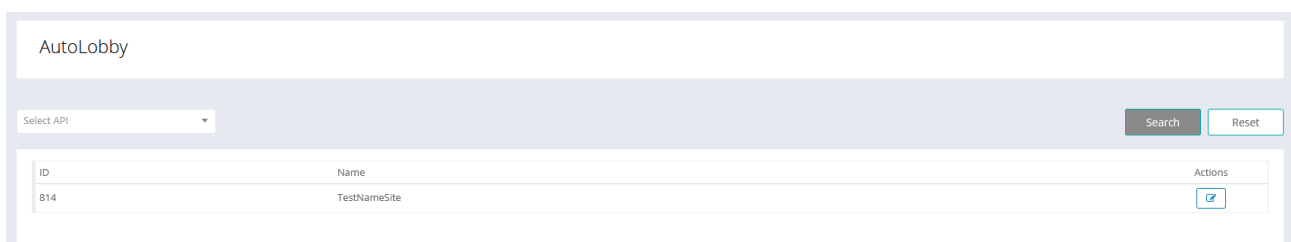
1. On success, JSON formatted data is returned
2. On error, and error code and description is returned, according to Appendix 1


JSON response example: {"1515604":{"en":1000},"1529345":{"en":432},"1555771":{"en":1,"enc":1},"1560667":{"ca":478781},"2132262":{"en":444,"enc":555},"2264176":{"en":20}}

7. Autolobby

7.1 Access in Fundist

A user with the access level "Network Operator" and "Site Operator" has access to the "AutoLobby" section. It is possible to configure various parameters for the lobby in this section.



ID	Name	Actions
814	TestNameSite	

The lobby is a kind of an HTML page with a catalog of games. The games are distributed by categories, providers and other blocks. It is possible to embed the lobby on a website page.

To create a lobby, you need to contact technical support, after that it will be possible to edit the lobby parameters in the Fundist system.

There are the following fields in the "Edit autolobby" modal window:

Api - the name of the Api connection and the lobby in the Fundist system.

Token - the access key to configure the lobby.

The CSS path is the path to the CSS file with styles for the lobby (example: <https://site.com/path/to/styles.css>).

Launch snippet - HTML code of the game launch button.

Settings - some settings parameters for the lobby are specified (for example: TopBlocks: New, TopBlockSize: 6).

Edit AutoLobby

Api

TestNameSite

Token

0d7c3d34b4325e11777d6b347b0f4408

CSS path

https://site.com/path/to/styles.css

Launch snippet

```

/**
 * @param gameId {string} - game id
 * @param demo {boolean} - is demo mode
 */
const testLaunchGameFn = (gameId, demo) => {
  alert(`Selected game with id ${gameId} in ${demo ? 'demo mode' : 'money mode'}`);
};

window.config = {
  launchGame: testLaunchGameFn,
}

```

Settings

Categories: tabs,
TopBlocks: New,
TopBlockSize: 6

Save

Close

7.2 How to launch the application of autolobby

You need to put the config with the application settings in the general context.

```

``js
window.config = {

```

```

country: 'lva',
language: 'en',
categories: {
  featured: ['22', '41', '38', '33'],
  menu: ['16', '37', '22', 'Sportsbook'],
},
topMerchantIDs: ['900', '967', '991', '997', '998'],
token: 'afc9338debcf98a0c5eb433a46d83799',
url: 'https://apiprod.fundist.org7',
startGameCallback: (gameId, demo) => {
  alert(`Selected game with id ${gameId} in ${demo ? 'demo mode' : 'money mode'}.
  \n* Implement this event handler on your side
  (window.config.startGameCallback)`);
},
customMenuCallback: (name) => {
  alert(`Custom menu "${name}" button clicked.
  \n* Implement this event handler on your side
  (window.config.customMenuCallback)`);
},
};
...

```

- country - user country code (3 letters)
- language - user language code (2 letters);
- categories.featured - featured categories displayed next to filters;
- categories.menu - IDs of categories displayed in the menu or custom menu labels;
- token - API access token;
- url - address of API;
- startGameCallback - a function that will be called when the game start button is clicked.
- customMenuCallback - a function that will be called when the custom menu is clicked.

In addition to the config, you need to connect the bundle with the application, create an element with the id "egamings-lobby" and set id="egamings-run-lobby" to the bundle script.

For tests, you can use test bundle: `<script id="egamings-run-lobby" defer="defer" src="https://test-autolobby.egamings.com/js/run.js"></script>` (see page 5);
 After updating the config, you need to reload the page.

7.3 Minimal Lobby Launch Template

```

<<html
<html lang="en">
<head>
<script>
window.config = {
country: 'lva',
language: 'en',
categories: {

```

⁷ <https://apiprod.fundist.org/>

```

featured: ['22', '41', '38', '33'],
menu: ['16', '37', '22', 'Sportsbook'],
},
topMerchantIDs: ['900', '967', '991', '997', '998'],
token: 'afc9338debcf98a0c5eb433a46d83799',
url: 'https://apiprod.fundist.org'8,
startGameCallback: (gameId, demo) => {
  alert(`Selected game with id ${gameId} in ${demo ? 'demo mode' : 'money mode'}.
  \n* Implement this event handler on your side
  (window.config.startGameCallback)`);
},
customMenuCallback: (name) => {
  alert(`Custom menu "${name}" button clicked.
  \n* Implement this event handler on your side
  (window.config.customMenuCallback)`);
},
};
</script>
<script id="egamings-run-lobby" defer="defer" src="https://test-autolobby.egamings.com/js/run.js"></
script (see page 5)>
</head>
<body>
<div id="egamings-lobby"></div>
</body>
</html>
...

```

7.4 Repainting

Repainting occurs through the config. The variables are the same as in the design system. The current palette is specified below:

```

```js
const sitePalette = {
 colors: {
 baseMain: '#FFFFFF',
 baseOpacity: 'rgba(147, 158, 183, 0.5)',
 buttonColor: '#7F7CFF',
 borderColor: '#5D6476',
 textColor: '#000000',
 main: '#7F7CFF',
 bg: '#1F1F26',
 bg100: '#525184',
 bg300: '#50566C',
 bg500: '#43485C',
 bg700: '#3D4150',
 secondary: '#1DC0C0',
 bgModal: '#2E313D',
 arrowColor: '#5D6476',
 }
}

```

---

<sup>8</sup> <https://apiprod.fundist.org/>

```
},
transition: {
 all: 'all .5s ease-out',
 duration: '.5s',
},
};
window.config = {
 // ...
 theme: sitePalette,
 // ...
};
...
```

## 8. Games Launch

### 8.1 Check for player country

There is a mandatory check for player country against the restricted country list on game launch. It's done in 2 steps:

1. Country is auto detected by using geoip library by UserIP value passed on API call.
2. If auto detected country from UserIP is valid, meaning, that it's not on restricted country list, then country from user account (on Fundist.ORG side) is checked. If player account country is valid, ie it's not on restricted country list – game launch process goes further.

NOTE: Please take into consideration, that in case player country is not passed or has unknown country code as a value on User/Add API call, country is auto detected by using geoip library from the RegistrationIP value. If player account is auto created on game launch (UserAutoCreate=1 is used), then the same rule applies to User/AuthHTML API calls: in case Country param is not passed or has unknown country code as a value, country is auto detected from the UserIP value.

### 8.2 Authorization with HTML fragment

User/AuthHTML is used for game launch, in response it returns HTML fragment.

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/User/AuthHTML/?

&Login=[LOGIN]

&Password=[PASSWORD]

&System=[SYSTEM]

&TID=[TID]

&Hash=[HASH]

&Page=[PAGE]

&UserIP=[USERIP]

{&Language=[LANGUAGE]}

{&Nick=[NICK]}

{&Timezone=[TIMEZONE]}

{&Demo=[DEMO]}

{&IsMobile=[ISMOBILE]}

{&ExtParam=[EXTPARAM]}

{&UserAutoCreate=[USERAUTOCREATE]}



```
{&Currency=[CURRENCY]}
{&Country=[COUNTRY]}
{&Referer=[REFERER]}
{&UniversalLaunch=[UNIVERSALLAUNCH]}
```

where:

[LOGIN] – user's login in API

[PASSWORD] – user's password in API

[SYSTEM] – the target merchant account. Should be checked via Game/FullList in "ProviderID" or "ID" value in "merchants" part.

[PAGE] – page code for redirection (can be found in the Game/List API call response )

[USERIP] – IP address of user – valid IPv4 (e.g. 12.34.56.78) or valid IPv6 address (for example 2001:0db8:85a3:0000:0000:8a2e:0370:7334). This value is used to identify country for the check against restricted country list for the specified gaming provider.

[HASH] – MD5 checksum of the following data:

User/AuthHTML/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[LOGIN]/[PASSWORD]/[SYSTEM]/[PWD]

Optional parameters:

[LANGUAGE] – language code (Appendix 3). User language is saved in database.

[NICK] – nick name, which prevents prompting on first entry

[TIMEZONE] – timezone shift in minutes relative to UTC. Valid range is from -720 till +720.

[DEMO] – either 1 or 0. Enables demo play mode for certain systems. It's ignored for lobby page.

[ISMOBILE] – either 1 or 0. Enables mobile device support. It's ignored for lobby page.

[EXTPARAM] – optional parameter, which will be present in OneWallet callbacks

[USERAUTOCREATE] – optional parameter, which indicates, wherever Fundist.ORG should create user account automatically. In case param has value 1, currency should be passed in order to create user account.

[CURRENCY] – The currency of user's account. It cannot be changed! Mandatory field, in case UserAutoCreate=1.

[COUNTRY] – ISO 2 or 3 symbol code.

[REFERER] – Lobby URL, meaning an URL, to which player should be redirected when player clicks/taps Home button on game UI. Mostly related to mobile version, as for desktop Home button usually is not displayed. Example: <https://www.google.com/doodles/>

[UNIVERSALLAUNCH] - either 1 or 0. Optional parameter, which will change the response format to json with the following parameters: 'gameHtml' : html\_fragment, 'gameScript' : js\_fragment.

NOTE: In case Country param is not passed or passed with unknown country code for Fundist.ORG system, country will be auto detected by UserIP value by using geoip library.

Some games could be launched in demo mode, for example for game play without user registration. Use the following parameter combination for demo play launch:

“Login=\$DemoUser\$&Demo=1&Password=Demo”

To identify which games could be launched in demo mode, you could check HasDemo field in the Game/List API call response for according game record.

**Server response:**

1. On success: 1,XXXXXX – where XXXXX is a HTML fragment to be inserted into website, which can contain commas (only the first comma should be processed).
2. On error, and error code and description is returned, according to Appendix 1

## 8.3 Kill authorization (end active sessions)

This request forcibly kills all active user sessions. It may not work for all systems.

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/User/KillAuth/?

&Login=[LOGIN]

&TID=[TID]

&Hash=[HASH]

{&ExtParam=[EXTPARAM]}

where:

[LOGIN] – user’s login in API

[HASH] – MD5 checksum of the following data:

User/KillAuth/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[LOGIN]/[PWD]

[EXTPARAM] – optional parameter. If set, kills only sessions associated with ExtParam value

**Server response:**

1. On success: 1
2. On error, and error code and description is returned, according to Appendix 1

## 9. Statistics

Based on a heavy demand of customers, we have redesigned and re-enabled statistics API. Please note that any legacy still working calls will be disabled as soon as the new API fully replaces the old one.

### 9.1 Universal Bets Summary

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/Stats/BetsSummary/?`

`&Date=[DATE]`

`&TID=[TID]`

`&Hash=[HASH]`

`{&Login=[LOGIN]}`

`{&System=[SYSTEM]}`

`{&Format=[FORMAT]}`

`{&AffiliateID=[AFFILIATEID]}`

where:

[DATE] – YYYY-MM-DD date for selection. Range selections are currently disabled.

[HASH] – MD5 checksum of the following data:

`Stats/BetsSummary/[CASINO_SERVER_IP]/[TID]/[KEY]/[DATE]/[PWD]`

#### Optional parameters:

[LOGIN] – apply filter based on user's login in API

[SYSTEM] – apply filter based on the merchant system

[FORMAT] – output format: "csv" or "json". Default: "csv"

[AFFILIATEID] – apply filter based on AffiliateID parameter passed to User/Add

#### Server response:

1. On success, either CSV or JSON formatted statistical data is returned
2. On error, and error code and description is returned, according to Appendix 1

#### CSV response:

First line: field names and their order

All other lines: user x system x date bets summary

**JSON response:**

```
{
 "fields": [field order list for data],
 "data": [
 [user x system x date bets summary],
 ...
]
}
```

**Possible fields:**

'user\_name' – user login as in API requests

'system\_id' – merchant's system

'currency\_name' – user's currency name (ISO code)

'bet\_total' – total amount of bets

'bet\_count' – total count of bets

'win\_total' – total amount of wins

'game\_desc' – game description

'gambling\_date' – date of the summary (added only in range selections)

## 9.2 Stats Bets

**NOTE:** Access to this interface is forbidden by default, please contact FUNDIST.ORG representative for according permission for API setup.

[URL] has a form of:

[https://\[SERVER\]/System/Api/\[KEY\]/Stats/Bets/?](https://[SERVER]/System/Api/[KEY]/Stats/Bets/?)

&Date=[DATE]

&TID=[TID]

&Hash=[HASH]

&System=[SYSTEM]

{&Login=[LOGIN]}

{&Format=[FORMAT]}

{&AffiliateID=[AFFILIATEID]}

{&Limit=[LIMIT]}

where:

[DATE] – YYYY-MM-DD date for selection. Range selections are currently disabled.

[SYSTEM] – apply filter based on the merchant system.

[HASH] – MD5 checksum of the following data:

Stats/Bets/[CASINO\_SERVER\_IP]/[TID]/ /[KEY]/[DATE]/[PWD]

#### **Optional parameters:**

[LOGIN] – apply filter based on user's login in API

[FORMAT] – output format: "csv" or "json". Default: "csv"

[AFFILIATEID] – apply filter based on AffiliateID parameter passed to User/Add

[LIMIT] – limit number of records in response. Default: 1000

#### **Server response:**

1. On success, either CSV or JSON formatted statistical data is returned

2. On error, and error code and description is returned, according to Appendix 1

CSV response:

First line: field names and their order

All other lines: date bet user x system x game x currency x bet x win x result x

#### **JSON response:**

```
{
 "fields": [field order list for data],
 "data": [
 [date bet user x system x game x currency x bet x win x result x],
 ...
]
}
```

#### **Possible fields:**

'date' – date of bet

'user' – user login as in API requests

'system\_id' – merchant's system

'game\_name' – game description

'currency\_name' – user's currency name (ISO code)

'amount\_bet' – amount of bet

'amount\_win' – total amount of wins

'casino\_result' – amount of profit a casino

## 9.3 Stats Detailed

**NOTE:** Access to this interface is forbidden by default, please contact FUNDIST.ORG representative for according permission for API setup.

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/Stats/Detailed/?

&Date=[DATE]

&TID=[TID]

&Hash=[HASH]

&System=[SYSTEM]

{&Login=[LOGIN]}

{&Format=[FORMAT]}

{&AffiliateID=[AFFILIATEID]}

{&Limit=[LIMIT]}

{&Offset}=[OFFSET]}

where:

[DATE] – YYYY-MM-DD date for selection. Range selections are currently disabled.

[SYSTEM] – apply filter based on the merchant system.

[HASH] – MD5 checksum of the following data:

Stats/Detailed/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[DATE]/[PWD]

### Optional parameters:

[LOGIN] – apply filter based on user's login in API

[FORMAT] – output format: "csv" or "json". Default: "csv"

[AFFILIATEID] – apply filter based on AffiliateID parameter passed to User/Add

[LIMIT] – limit number of records in response. Default: 1000

[OFFSET] – offset a results of the response. Default: 0

### Server response:

1. On success, either CSV or JSON formatted statistical data is returned

2. On error, and error code and description is returned, according to Appendix 1

CSV response:

First line: field names and their order

All other lines: user x system x date action game id game x currency x amount x action\_id x action\_type x

### JSON response:

```
{
 "fields": [field order list for data],
 "data": [
 [user x system x date action game id game x currency x amount x action_id x action_type x],
 ...
]
}
```

**Possible fields:**

'user' – user login as in API requests

'system\_id' – merchant's system

'date' – date of action

'game\_id' – id of game

'game\_name' – game description

'currency\_name' – user's currency name (ISO code)

'amount' – amount of action

'action\_id' – id of action

'action\_type' – type of action

## 9.4 Game Detailed

**NOTE:** Access to this interface is forbidden by default, please contact FUNDIST.ORG representative for according permission for API setup.

[URL] has a form of:

[https://\[SERVER\]/System/Api/\[KEY\]/Stats/GameDetails/?](https://[SERVER]/System/Api/[KEY]/Stats/GameDetails/?)

&GameID=[GAMEID]

&TID=[TID] &Hash=[HASH]

where:

[GAMEID] – ID a game.

[HASH] – MD5 checksum of the following data:

Stats/GameDetails/[CASINO\_SERVER\_IP]/[TID]/ /[KEY]/[GAMEID]/[PWD]

**Server response:**

1. On success, JSON formatted data is returned
2. On error, and error code and description is returned, according to Appendix 1

**JSON response:**

```

{
 "i_gameid": "329049",
 "userid": "292",
 "currency": "EUR",
 "system_id": "998",
 "start_time": "2017-05-10 16:26:29",
 "end_time": "2017-05-10 16:26:35",
 "i_gamedesc": "casinoholdem",
 "i_extparam": "",
 "actions": [
 {
 "i_actionid": "14bcb3debf040e91ab3a6ca7",
 "type": "bet",
 "action_amount": "10.00",
 "action_time": "2017-05-10 16:58:08"
 },
 . . .
],
 "details": {
 "175228": {
 "tableId": "HoldemTableXXX",
 "gameId": "14bcb3debf040e91ab3a6ca7",
 "gameType": "casinoholdem",
 "timestamp": "Mon May 08 18:09:22 976 UTC 2017",
 "player": {
 "euID": "192_292",
 "nickname": "nickname",
 "actionState": "final",
 "playerAction": {
 "actionType": "ante",
 "betCode": "HLDM_AnteBet",
 "desc": "Ante Bet",
 "stakedAmount": "100"
 }
 }
 }
 }
}

```



```

},
...
},
}

```

**Possible fields:**

'i\_gameid' – ID a game  
 'userid' – user login as in API requests  
 'currency' – game currency name (ISO code)  
 'system\_id' – merchant's system  
 'start\_time' – date of start game  
 'end\_time' – date of end game  
 'i\_gamedesc' – game description  
 'i\_extparam' – game external params  
 'action\_amount' – amount of action  
 'i\_actionid' – external id of action  
 'type' – type of action  
 'action\_time' – time of action  
 'action\_amount\_real' – real part of bet amount  
 'action\_amount\_bonus' – bonus part of bet amount

## 9.5 Loyalty payments

[URL] has a form of:

[https://\[SERVER\]/System/Api/\[KEY\]/Loyalty/Payment?](https://[SERVER]/System/Api/[KEY]/Loyalty/Payment?)

&Login=[LOGIN]

&Amount=[AMOUNT]

&TID=[TID]

&Hash=[HASH]

&UserIP=[USERIP]

{&BonusCode=[BONUSCODE]}

{&BonusID=[BONUSID]}

{&System=[SYSTEM]}

where:

[LOGIN] – user's login in API

[SYSTEM] – the target merchant account

[AMOUNT] – Amount of transaction. Positive amount – cash in, negative amount – cash out.

[TID] – Unique transaction ID (for whole history of requests, 32 chars max)

[HASH] – MD5 checksum of the following data:

Loyalty/Payment/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[SYSTEM]/[AMOUNT]/[LOGIN]/[PWD]

[BONUSCODE] – promo code which will activate according to the payment

[BONUSID] – bonus id which will activate according to the payment

[USERIP] – IP address of user.

#### **Server response:**

1. On success, "1,Ok" returned.
2. On error, error code and description is returned, according to Appendix 1

## **9.6 Bonus select**

[URL] has a form of:

[https://\[SERVER\]/System/Api/\[KEY\]/Loyalty/Bonus/Select?](https://[SERVER]/System/Api/[KEY]/Loyalty/Bonus/Select?)

&IDBonus=[IDBONUS]

&Status=[STATUS]

&Login=[LOGIN]

&TID=[TID]

&Hash=[HASH]

&UserIP=[USERIP]

{&PromoCode=[PROMOCODE]}

where:

[IDBONUS] – Bonus id

[STATUS] – Status of bonus:

0 - Disable bonus,

1 - Select bonus,

2 - Take inventory bonus.

[LOGIN] – user's login in API

[TID] – Unique transaction ID (for whole history of requests, 32 chars max)

[HASH] – MD5 checksum of the following data:

Loyalty/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[LOGIN]/[PWD]

[USERIP] – IP address of user.

[PROMOCODE] - promo code which will allow player to use a bonus

**Server response:**

1. On success, "1,{"Status":1}" returned.
2. On error, error code and description is returned, according to Appendix 1 or error message of bonus selection.

## 9.7 Bonus cancel

[URL] has a form of:

[https://\[SERVER\]/System/Api/\[KEY\]/Loyalty/Bonus/Cancel?](https://[SERVER]/System/Api/[KEY]/Loyalty/Bonus/Cancel?)

&IDBonus=[IDBONUS]

&Login=[LOGIN]

&TID=[TID]

&Hash=[HASH]

&UserIP=[USERIP]

where:

[IDBONUS] – Bonus id

[LOGIN] – user's login in API

[TID] – Unique transaction ID (for whole history of requests, 32 chars max)

[HASH] – MD5 checksum of the following data:

Loyalty/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[LOGIN]/[PWD]

[USERIP] – IP address of user.

**Server response:**

1. On success, "1,{"Status":1}" returned.
2. On error, error code and description is returned, according to Appendix 1 or error message of bonus cancelling.

## 9.8 Subscribe tournament

[URL] has a form of:

[https://\[SERVER\]/System/Api/\[KEY\]/Loyalty/Tournament/Select?](https://[SERVER]/System/Api/[KEY]/Loyalty/Tournament/Select?)

&IDTournament=[IDTournament]

&Status=[STATUS]

```
&Login=[LOGIN]
&TID=[TID]
&Hash=[HASH]
{&UserIP=[USERIP]}
{&Balance=[BALANCE]}
```

where:

[IDTournament] – Tournament id

[LOGIN] – User's login in API

[STATUS] – Status of tournament:

0 - Disable tournament,

1 - Select tournament.

[BALANCE] – User's balance, by default it will be requested via OW protocol

[TID] – Unique transaction ID (for whole history of requests, 32 chars max)

[HASH] – MD5 checksum of the following data:

Loyalty/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[LOGIN]/[PWD]

[USERIP] – IP address of user.

#### Server response:

1. On success, "1,{"Status":1}" returned.
2. On error, error code and description is returned, according to Appendix 1 or error message of tournament subscriptiong.

## 9.9 Order store item

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/Loyalty/Store/Buy?

&IDItem=[IDITEM]

&Login=[LOGIN]

&TID=[TID]

&Hash=[HASH]

{&UserIP=[USERIP]}

{&Balance=[BALANCE]}

{&Country=[COUNTRY]}

{&Quantity=[QUANTITY]}

where:

[IDITEM] – Store item id

[LOGIN] – User's login in API

[BALANCE] – User's balance, by default it will be requested via OW protocol

[COUNTRY] - User's country

[QUANTITY] - Quantity of items, default 1

[TID] – Unique transaction ID (for whole history of requests, 32 chars max)

[HASH] – MD5 checksum of the following data:

Loyalty/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[LOGIN]/[PWD]

[USERIP] – IP address of user.

**Server response:**

1. On success, "1,{json string of order data}" returned. Description JSON see 5.6 Order store item in Loyalty Api Basic Document.
2. On error, error code and description is returned, according to Appendix 1 or error message of store order.

## 10. Error handling

### 10.1 Retrieve response of previously executed request

This functionality can be used to workaround connection problems.

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/Response/Get/?&TID=[TID]`

**Server response:**

- The original response, if found
- NOT\_FOUND, if not found
- KEY\_ERROR, if [KEY] is invalid
- IP\_ERROR, if requesting IP is not allowed for [KEY]

### 10.2 Retrieve information of previously executed request

The request is identical to Response/Get, but provides additional information for better robustness of automatic error handling and recovery. Response/Get has a potential race condition, if duplicated [TID] requests are received.

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/Response/Info/?&TID=[TID]`

**Server response:**

If request [TID] is found then response consists of several lines separated by newline ('\n') character:

1. Request (Balance/Get, Balance/Set, etc.)
2. Arguments (?&=....)
3. Response code
4. Full response
5. Date and time of request
6. IP of request

**On error:**

- NOT\_FOUND, if not found
- KEY\_ERROR, if [KEY] is invalid
- IP\_ERROR, if requesting IP is not allowed for [KEY]



## 11. Providers additional info

### 11.1 Get live providers table info

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/Providers/TablesInfo/?

&Provider=[Provider]

&TID=[TID]

&Hash=[HASH]

{&Currencies=[CURRENCIES]}

where:

[CURRENCIES] - one ISO 4217 currency code or set of ISO 4217 currency codes separated by commas - EUR,USD,...

[HASH] – MD5 checksum of the following data:

Providers/TablesInfo/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[PWD]

#### Server response:

1. On success, JSON formatted data is returned
2. On error, and error code and description is returned, according to Appendix 1

#### JSON response:

```
[
 "TableID":{
 "name" : "Table name",
 "limits" : {
 "EUR" : {
 "symbol" : "EUR",
 "min":1,
 "max":1000
 },
 ...
 },
 "availability" : {
 "open":true,
```



```

"operationHours":{
 "type":"FullTime"
},
"resultsHistory":["33","25","13","10","3","3","6","12","14","1"]
},
...
]

```

Note: resultsHistory presents if supported by provider.

## 11.2 Get Amusnet / EGT Digital jackpots widget\*

*\*Disabled by default. For enabling contact Integration Team.*

Kindly note that this request is available for Amusnet and EGT Digital only.

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/Game/InternalJackpots?`

`&TID=[TID]`

`&Hash=[HASH]`

`&IDMerchant=[IDMERCHANT]` - currently ignored

where:

[IDMERCHANT] - merchant system number

[HASH] – MD5 checksum of the following data: Game/InternalJackpots/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[PWD]

### Server response:

1. On success, JSON formatted data is returned
2. On error, and error code and description is returned, according to Appendix 1 or an empty field

### JSON response example:

```

"954": {
 "currency": "EUR",
 "currentLevell": 1114,
 "winsLevell": 26039,
 "largestWinLevell": 9778,
 "largestWinDateLevell": "Jun 2, 2023 10:19:59 AM",
 "largestWinUserLevell": "UserName1",
 "lastWinLevell": 1707,
 "lastWinDateLevell": "Jun 20, 2023 11:46:43 AM",
 "lastWinUserLevell": "UserName2",
 "topMonthlyWinnersLevell": [{
 "winUser": "UserName1",
 "winAmount": 9778,
 "winDate": "Jun 2, 2023 10:19:59 AM"
 }
]

```

```

}, {
 "winUser": "UserName3",
 "winAmount": 7615,
 "winDate": "Jun 16, 2023 11:59:06 AM"
},
],
"topYearlyWinnersLevel": [{
 "winUser": "UserName1",
 "winAmount": 9778,
 "winDate": "Jun 2, 2023 10:19:59 AM"
}, {
 "winUser": "UserName4",
 "winAmount": 8168,
 "winDate": "May 29, 2023 5:40:44 PM"
},
], etc

```

where:

"currency" - Amusnet configuration's main currency. If conversion is required, please do it yourself on your side, as the provider cannot provide such an option.

Level I – Clubs, Level II – Diamonds, Level III – Hearts, Level IV – Spades

"currentLevel" – Value (contribution) that the respective level has at this moment. This is the value that will be visualized for "Clubs" level on the Jackpot banner.

"winsLevel" – Number of wins that has been recorded for the respective Jackpot level.

"largestWinLevel" – The biggest amount that has been won for the respective level.

"largestWinUserLevel" – The particular user that have won the biggest amount on the respective JP level.

"lastWinLevel" – Last win (last drop) initiated for the respective JP level.

"lastWinDateLevel" – Last time that the respective JP level has been won.

"lastWinUserLevel" – ID, or display name of the last user that has won the respective JP level.

"topMonthlyWinnersLevel" – Players with the largest win amounts on monthly basis.

"topYearlyWinnersLevel" – Players with the largest win amounts on yearly basis.

## 11.3 Get jackpots list

This API call is intended to get game jackpot amounts for those providers who support this feature (may be checked with integration team)

[URL] has a form of:

https://[SERVER]/System/Api/[KEY]/WLCGames/Jackpots?

&TID=[TID]

&Hash=[HASH]

&IDMerchant=[IDMerchant]

&Currency=[Currency]

where

[KEY] – API access key

[PWD] – API access password

[IDMerchant] - Optional parameter. API merchant id

[Currency] - Optional parameter. short code of currency (EUR, USD, RUB...)

[HASH] – MD5 checksum of the following data:

WLCGames/[CASINO\_SERVER\_IP]/[TID]/[KEY]/[PWD]

### Server response:

On success: JSON data of available jackpots

On error, and error code and description is returned, according to Appendix 1

## 11.4 Get Min/Max bet (PGSoft only)

This API call is intended to get game min/max amounts for the providers that support this feature (may be checked with integration team)

[URL] has a form of:

`https://[SERVER]/System/Api/[KEY]/WLCGames/MinMaxBet?`

`&TID=[TID]`

`&Hash=[HASH]`

`&IDMerchant=[IDMerchant]`

`&Currency=[Currency]`

where

[KEY] – API access key

[PWD] – API access password

[IDMerchant] - API merchant id

[Currency] - Optional parameter. short code of currency (EUR, USD, RUB...). When is NULL, return empty response

[HASH] – MD5 checksum of the following data:

`WLCGames/[CASINO_SERVER_IP]/[TID]/[KEY]/[PWD]`

### Server response:

On success: JSON data of available min/max bets

On error, and error code and description is returned, according to Appendix 1

### Response example for Provide PgSoft:

```
{
 "data": [
 {
 "gameLegalBetAmounts": [
 {
 "gameId": 1,
 "gameTypeId": 0,
 "legalBetAmounts": [
 {
 "coinSize": 0.01,
 "betMultiplier": 1,
 "baseBet": 30,
 "betAmount": 0.3
 }
],
 "coinSize": 0.01,
```

```

 "betMultiplier": 2,
 "baseBet": 30,
 "betAmount": 0.6
 },
 {
 "coinSize": 0.01,
 "betMultiplier": 3,
 "baseBet": 30,
 "betAmount": 0.9
 },
 {
 "coinSize": 0.01,
 "betMultiplier": 4,
 "baseBet": 30,
 "betAmount": 1.2
 },
 {
 "coinSize": 0.01,
 "betMultiplier": 5,
 "baseBet": 30,
 "betAmount": 1.5
 },
 {
 "coinSize": 0.01,
 "betMultiplier": 6,
 "baseBet": 30,
 "betAmount": 1.8
 },
 {
 "coinSize": 0.01,
 "betMultiplier": 7,
 "baseBet": 30,
 "betAmount": 2.1
 },
 {
 "coinSize": 0.01,
 "betMultiplier": 8,
 "baseBet": 30,
 "betAmount": 2.4
 },
 {
 "coinSize": 0.01,
 "betMultiplier": 9,
 "baseBet": 30,
 "betAmount": 2.7
 },
 {
 "coinSize": 0.01,
 "betMultiplier": 10,
 "baseBet": 30,
 "betAmount": 3.0
 },
 {

```

```

 "coinSize": 0.05,
 "betMultiplier": 1,
 "baseBet": 30,
 "betAmount": 1.5
 },
]
}
],
"gameId": 1,
"gameName": "Honey Trap of Diao Chan",
"gameCode": "diaochan",
"status": 1,
"releaseStatus": 1,
"isSupportFreeGame": true,
"category": 1
},
.....

```

On response:

"gameId" - Provider game ID

"gameName" Game name

"gameCode" Game launch code

"status": status of game (1 - enabled, 0 - disabled)

"releaseStatus" On release status (1 - enabled, 0 - disabled) ,

"isSupportFreeGame" Support free rounds of game (true - support, false - not support),

"category" Game category

gameLegalBetAmounts Item (Field names corresponds its meanings) :

"coinSize": 0.01,

"betMultiplier": 1,

"baseBet": 30,

"betAmount": 1.5

## 11.5 Get Sport bets

API to get betting data in BetRadar

### Request

[URL] has a form of:

[https://\[SERVER\]/System/Api/\[KEY\]/History/GetSportBets](https://[SERVER]/System/Api/[KEY]/History/GetSportBets)

?TID=[TID]

&Hash=[HASH]

&merchantId=[SYSTEM]

... & parameters for BetRadar or Digitain ...

Please note that for GetSportBets only Betradar (958) or Digitain (972) SYSTEM can be sent, if other will be sent then the system will change it to the default which is currently Betradar (958).

```
$hash = md5(
 'HistoryRouter/GetSportBets' .
 '/0.0.0.0/' .
 $transactionId .
 '/' .
 $ApiKey .
 '/' .
 $ApiPass
);
```

Possible parameters for filtering both for Betradar or Digitain:

```
lang;
userId;
dateStart;
dateEnd;
type;
status;
result;
limit;
offset;
id;
```

order;

### Response example if you use Betradar:

```
{
 "code": 200,
 "status": "success",
 "data": [
 {
 "id": "57070",
 "type": "single",
 "systemBet": 0,
 "status": "closed",
 "result": "win",
 "amount": 10,
 "possibleWin": 10.300000000000001,
 "payout": 10.300000000000001,
 "createdAt": 1698650575091,
 "odds": 1.03,
 "events": [
 {
 "id": "22845",
 "result": "win",
 "odds": 1.03,
 "entityType": "match",
 "outcomeName": "Gavrila, Oana",
 "marketName": "Winner",
 "teamHomeName": "Marthiens, Karen",
 "teamAwayName": "Gavrila, Oana",
 "seasonName": null,
 "tournamentName": "ITF Greece 05A, Women Singles",
 "categoryName": "ITF Women",
 "sportName": "Tennis",
 "eventStartDate": "2023-10-30T07:00:00.000Z",
 "teamHomeScore": 0,
 "teamAwayScore": 2
 }
]
 },
 {
 "id": "57173",
 "type": "single",
 "systemBet": 0,
 "status": "accepted",
 "result": null,
 "amount": 10,
 "possibleWin": 17.699999999999999,
 "payout": null,
 "createdAt": 1700140664282,
 "odds": 1.77,
 "events": [
```



```

 {
 "id": "22975",
 "result": null,
 "odds": 1.77,
 "entityType": "match",
 "outcomeName": "Cash J / Stevens B",
 "marketName": "Winner",
 "teamHomeName": "Cash J / Stevens B",
 "teamAwayName": "Balaji N S / Begemann A",
 "seasonName": null,
 "tournamentName": "ATP Challenger Danderyd, Sweden Men Doubles",
 "categoryName": "Challenger",
 "sportName": "Tennis",
 "eventStartDate": "2023-11-16T13:00:00.000Z",
 "teamHomeScore": 1,
 "teamAwayScore": 0
 }
]
}

```

### Response example if you use Digitain:

```

{
 "ActionData": {
 "Token": "10312451-f8865*****966a0ac",
 "CurrencyId": "EUR",
 "OrderNumber": 2000432235,
 "GameId": 2,
 "TransactionId": 3635713421,
 "Info": "Place Bet.",
 "DeviceTypeId": 2,
 "TypeId": 1,
 "BetState": 1,
 "PossibleWin": 3.76,
 "BetTaxAmount": 0,
 "BonusId": 0,
 "TrackingId": "2e2d3c40-*****-c4a548e2d571",
 "OperationItems": [
 {
 "Token": "10312451-f8865*****966a0ac",
 "CurrencyId": "EUR",
 "OrderNumber": 2000432235,
 "GameId": 2,
 "TransactionId": 3635713421,
 "Info": "Place Bet.",
 "DeviceTypeId": 2,
 "TypeId": 1,
 "BetState": 1,
 "PossibleWin": 3.76,
 "BetTaxAmount": 0,
 "BonusId": 0,
 "TrackingId": "2e2d3c40-4065-4fa9-b7e8-c4a548e2d571",
 "OperationItems": [],
 "BetCommission": 0,
 "ViewTypeId": 1,
 "Order": {
 "lottreyBonusMessage": ""
 },
 "OrderBetType": 2,
 "Bets": [
 {
 "BetTypeId": 1,
 "SystemIndex": 0,
 "Factor": 1.88,
 "BetStakes": [
 {
 "Factor": 1.88,
 "EventDate": "2025-09-15T06:40:00Z",
 "EventName": "Ararat Armenia - Ararat Moscow[0:0 (0:0, 0:0)] [Not Started] {Football # Italy # Italy. Serie A}",
 "EventNameOnly": "Ararat Armenia - Ararat Moscow",
 "EventScore": "0 : 0",
 "FullStake": "Result: X",
 "StatusName": "New",
 "Status": 1,
 "StakeId": 3231745887,
 "EventDateStr": "9/15/2025 6:40:00 AM",
 "EventDateTimeStamp": 1757918400000,
 "CreateEventDate": 1757918441540,
 "SportID": 1,
 "SportName": "Football",
 "Score": "[0:0 (0:0, 0:0)] [Not Started]",
 "TournamentID": 4484,
 "TournamentName": "Italy. Serie A",
 "EventShortID": 10628,
 "EventId": 5652392,
 "StakeName": "X",
 "StakeTypeName": "Result",
 "PeriodName": "",
 "Argument": null,
 "IsLive": true,
 "ShowSign": false,
 "StakeTypeID": 1,
 "EventNumber": 5652392,
 "CategoryName": "Italy",
 "CategoryID": 1287,
 "IsCashout": false,
 "Teams": [
 {
 "Side": 1,
 "Id": 254118,
 "Name": "Ararat Armenia",
 "ShortName": null,
 "Side": 2,
 "Id": 177626,
 "Name": "Ararat Moscow",
 "ShortName": null
]
 },
 "CategoryShortName": "ITA",
 "IsOutright": false,
 "IsSpecialEvent": false,
 "ConfirmResult": 0,
 "BoostType": 0
 }
],
 "BetAmount": 2,
 "BonusFactor": 1,
 "MaxWinAmount": 3.76,
 "WinAmount": 0,
 "IsWinner": false,
 "MaxAllFactor": 1.88,
 "CashbackAmount": 0,
 "IsGiveCashBack": false,
 "IsCashout": false,
 "FullName": "Bet type: Online Single",
 "BonusType": 0,
 "BonusTypeName": null,
 "DayMBetID": 0
 }
],
 "CheckNumber": 0,
 "WinAmount": 0,
 "CashoutAmount": 1,
 "IsWinner": false,
 "Status": 1,
 "StatusName": "New",
 "IsOutOfRisk": false,
 "StampNumber": -1,
 "OrderNumber": 2000432235,
 "FillDate": "2025-10-20T13:12:16.4732692+04:00",
 "FillDateStr": "10/20/2025 1:12:16 PM",
 "IsRePrintRequested": false,
 "IsPrintAllowed": false,
 "OrderAmount": 2,
 "MaxWinAmount": 3.76,
 "AlphabeticCode": null,
 "BetShopName": "",
 "SystemBetMinWinAmount": 0,
 "IsLive": true,
 "FullName": null,
 "IsUnlimitBet": false
 }
]
 }
}

```

```

{"IsInternetBet":true,"Multiplyfactor":1,"OrderCode":"","IsCashBack":false,"Time":"1:12
PM","SportLotoNumbers":"","IsBetBuilder":false},
{"BetMinLimit":0.1,"BetMaxLimit":2000,"IpAddress":"RU 188.130.240.79","IsAutoBet":false,
"PartnerBetId":null,
"TerminalId":0,"PromotionType":0,"PinCode":51947595,"PartnerId":153,"TimeStamp":1760965943
,"Signature":"363cb65*****303003e37"}],{"BetCommission":0,"ViewTypeID":1,"Order":
{"lottreyBonusMessage":"","OrderBetType":2,"Bets":
[{"BetTypeId":1,"SystemIndex":0,"Factor":1.88,"BetStakes":[{"Factor":1.88,"EventDate":
"2025-09-15T06:40:00Z","EventName":"Ararat Armenia - Ararat Moscow[0:0 (0:0, 0:0)] [Not
Started] {Football # Italy # Italy. Serie A}","EventNameOnly":"Ararat Armenia - Ararat
Moscow","EventScore":"0 : 0","FullStake":{"Result:X"},"StatusName":"New",
"Status":1,"StakeId":3231745887,"EventDateStr":"9/15/2025 6:40:00 AM",
"EventDateTimeStamp":1757918400000,"CreateEventDate":1757918441540,"SportID":1,"SportName":
"Football","Score":"[0:0 (0:0, 0:0)] [Not Started] ","
"TournamentID":4484,"TournamentName":"Italy. Serie A",
"EventShortID":10628,"EventId":5652392,"StakeName":"X","StakeTypeName":"Result",
"PeriodName":"","Argument":null,"IsLive":true,"ShowSign":false,
"StakeTypeID":1,"EventNumber":5652392,"CategoryName":"Italy",
"CategoryID":1287,"IsCashout":false,"Teams":[{"Side":1,"Id":254118,"Name":"Ararat
Armenia","ShortName":null},{ "Side":2,"Id":177626,"Name":"Ararat Moscow",
"ShortName":null}], "CategoryShortName":"ITA","IsOutright":false,"IsSpecialEvent":false,
"ConfirmResult":0,"BoostType":0}],
"BetAmount":2,"BonusFactor":1,"MaxWinAmount":3.76,"WinAmount":0,"IsWinner":false,
"MaxAllFactor":1.88,"CashbackAmount":0,"IsGiveCashBack":false,"IsCashout":false,
"FullName":"Bet type: Online Single","BonusType":0,"BonusTypeName":null,"DayMBetID":0}],
"CheckNumber":0,"WinAmount":0,"CashoutAmount":1,"IsWinner":false,
"Status":1,"StatusName":"New","IsOutOfRisk":false,
"StampNumber":-1,"OrderNumber":2000432235,"FillDate":"2025-10-20T13:12:16.4732692+04:00",
"FillDateStr":"10/20/2025 1:12:16 PM","IsRePrintRequested":false,"IsPrintAllowed":false,
"OrderAmount":2,"MaxWinAmount":3.76,"AlphabeticCode":"","BetShopName":"","
"SystemBetMinWinAmount":0,"IsLive":true,"FullName":null,"IsUnlimitBet":false,
"IsInternetBet":true,"Multiplyfactor":1,"OrderCode":"","IsCashBack":false,"Time":"1:12
PM","SportLotoNumbers":"","IsBetBuilder":false},
{"BetMinLimit":0.1,"BetMaxLimit":2000,"IpAddress":"RU 188.130.240.79","IsAutoBet":false,
"PartnerBetId":null,
"TerminalId":0,"PromotionType":0,"PinCode":51947595,"PartnerId":153,"TimeStamp":1760965943
,"Signature":"363cb6*****3e37"}],
 "InternalGameId": "77123",
 "ExternalGameId": "2000432235",
 "ActionType": "bet",
 "ActionExternalId": "3635713421"
}

```

## 12. Frequently asked questions (FAQ)

**1. Question:** I constantly get “30,Transaction ID failed value” - what I am doing wrong?

**Answer:** Each API call should have unique TID value, basically it's unique API request ID, please make sure your backend passes unique TID value each time.

Another possible problem API call URL : it's important that API call parameters follow relative path by “?&” character combination, not just standard “?”.

**2. Question:** I have changed server IP, it has been whitelisted for API access, but I get “15,Wrong hash authorization” - what's wrong?

**Answer:** Please make sure, you use the correct server IP, when Hash param is calculated – server IP value should be the same as real external server IP, from which actual API call is made.

If it's not possible to determine external server IP at Hash param calculation time, please use 0.0.0.0 for Hash generation as IP value.

**3. Question:** How can I get game's PageCode?

**Answer:** You can get it from “Game/List/?” or “Game/FullList/?” commands (see chapter 3 for more info).

**4. Question:** How often do I have to update Game Catalogue?

**Answer:** We recommend you to do it at least one time a day.

**5. Question:** I get “0” on my balance request. What should I do?

**Answer:** I need to check HMAC calculation in answer on balance request

**6. Question:** What I need to use in SystemID (MerchantID) value?

**Answer:** You need to check "ProviderID" or "ID" in "merchants" part of Game/FullList

**7. Question:** What is the difference between “Game/List/?” and “Game/FullList/?”

**Answer:** “Game/FullList/?” request gives you more game details

**8. Question:** How can I launch games in DemoMode?

**Answer:** You need to include parameters “Login=\$DemoUser\$&Demo=1&Password=Demo” into your request.

**9. Question:** What games have DemoMode?

**Answer:** You can check it in parameter “HasDemo” into “Game/List/?” or “Game/FullList/?” request

**10. Question:** How can I launch game's mobile version?

**Answer:** You need to use parameter IsMobile=1 + game's mobile PageCode into your request.

## 13. Appendices

## Appendix 1. Error codes

Error code	Description
0,Empty list	Failed to get list of categories
0,The entered characters are invalid	Prohibited characters were provided
0,The field "Date of birth" must be filled in	The "Date of birth" field is empty
0,No requested merchant	There is no merchant in the request
11,Wrong authorization	Invalid API [KEY] specified
12,Wrong authorization ip	Request IP is not allowed for specified API [KEY]
13,Wrong action	Wrong action is requested
13,Wrong action,User disabled by higher level operator	Unable to enable user, user has been disabled by higher level operator
13,Wrong action,User enabled	User already enabled
13,Wrong action,User disabled	User already disabled
13,Wrong Action,Language	Language is not supported
13,Wrong params	Incorrect game sorting options

Error code	Description
14,Wrong incoming params	Wrong incoming arguments are specified
14,Wrong incoming params,Date	Wrong date is specified
14,Wrong incoming params,Format	Wrong output format is specified
14,Either system id or player login should be provided	The system id and player login fields are missing
14,Wrong incoming params: missing System param	The system param is missing
14,Invalid registration IP	Wrong registration IP
14,Invalid Country	Wrong country code is provided
14,Invalid country state	Wrong country state code is provided
14,Invalid Country Iso Code by IP	Unable to get country ISO code by registration IP
15,Wrong authorization hash	Request hash is not properly calculated
16,Registration is restricted in the site settings	Registration is prohibited by site settings
16,User update error, [desc]	Error updating user information with description
16,User creation error,Race request	Race condition error while creating user

Error code	Description
16,User creation error, [desc]	Error creating user with description
17,User not found	Unknown user login
17,User is disabled	User disabled
17,There is no self-exclusion	Self-exclusion disabled
18,Wrong merchant access	Unsupported action for specified merchant
18,IDMerchant is Mandatory	IDMerchant param is missing
19,Transaction fail(amount), [desc]	The transaction contains an incorrect amount
19,Transaction fail, [desc]	Internal transaction failure with description
20,Merchant transaction fail,[desc]	External transaction failure with description
21,Can't get balance	Failed to get balance from merchant
22,Wrong user password	Wrong user password is supplied
23,User disabled	Request for disabled/blocked user
23,User waiting verification	The user has not been verified yet
24,Redirect error	Redirection error
24,Session limit expired	The user's session limit has expired

Error code	Description
24,Phone verification failure	Wrong phone number provided
24,Merchant error,Merchant not found	Wrong merchant ID provided
24,Redirect error,ExclusionByTag	User excluded by tag
24,self-exclusion, [date]	User self-excluded (expiration date is displayed if specified)
24,Redirect error,Currency not supported	The currency is not supported by merchant
24,Redirect error,Demo not supported	The demo version is not supported in this game
24,Redirect error,Branded games are not enabled	Branded games are forbidden
24,Redirect error,Restricted country	Country is not allowed by merchant
24,Redirect error,Unable to launch game	Launching games is prohibited
24,Redirect error, DirectAuth not supported (use AuthHTML)	DirectAuth is not supported, use AuthHTML instead
24,Redirect error,Stall suspended	The site is temporarily unavailable
26,User creation error	Failure to create a user
27,Your stall disabled	Request to disabled stall



Error code	Description
28, Your net disabled	Request to disabled network
29, Your API disabled	Request to disabled API setup
30, Transactions ID failed value	Duplicate [TID]
31, Wrong currency	Either specified currency is not enabled for stall or user's currency differs.
32, Wrong start or end dates	Start and end dates are not in specified format
33, This site is on maintenance	System is temporary unavailable
35, User currency could not be changed	User currency cannot be changed
36, Wrong user nickname	This nickname is blacklisted
48, Missing or invalid UserIP	Missing or invalid value provided for UserIP parameter.
49, User creation error, Email busy	Request creating user with used email. For one unique email kiosk.
50, Stall suspended	The site is temporarily unavailable
51, Creating games list lock	Occurs when sending a second request to receive a gamelist, if the initial request has not yet been completed.
666, Manual Verify	This situation should be manually processed by technical support.
NN,...	Any other code is an error

## Appendix 2. Currency codes

Currency code	Country	Currency name
AED	United Arab Emirates	United Arab Emirates dirham
AMD	Armenia	Armenian dram
ARS	Argentina	Argentine peso
AUD	Australia	Australian dollar
AZN	Azerbaijan	Azerbaijani manat
BGN	Bulgaria	Bulgarian lev
BRL	Brazil	Brazilian real
BYN	Belarus	Belarusian ruble
CAD	Canada	Canadian dollar
CHF	Switzerland	Swiss franc
CLP	Chile	Chilean peso
CNY	China	Renminbi, Chinese yuan
COP	Colombia	Colombian peso
CZK	Czech Republic	Czech koruna
DKK	Denmark	Danish krone

Currency code	Country	Currency name
EUR	European Union	Euro
GBP	Great Britain	Pound sterling
GEL	Georgia	Georgian lari
GHS	Ghana	Ghanaian cedi
HKD	China (Hong Kong)	Hong Kong dollar
HUF	Hungary	Hungarian forint
ILS	Israel	Israeli new shekel
INR	India	Indian rupee
IQD	Iraq	Iraqi dinar
IRR	Iran	Iranian rial
JPY	Japan	Japanese yen
KES	Kenya	Kenyan shilling
KGS	Kyrgyzstan	Kyrgyzstani som
KRW	South Korea	South Korean won
KZT	Kazakhstan	Kazakhstani tenge
MBC	Worldwide	Millibitcoin
MDL	Moldova	Moldovan leu
MNT	Mongolia	Mongolian tögrög

Currency code	Country	Currency name
MXN	Mexico	Mexican peso
MYR	Malaysia	Malaysian ringgit
NGN	Nigeria	Nigerian naira
NOK	Norway	Norwegian krone
NZD	New Zealand	New Zealand dollar
PEN	Peru	Peruvian Sol
PLN	Poland	Polish złoty
RON	Romania	Romanian leu
RUB	Russia	Russian ruble
SEK	Sweden	Swedish krona/kronor
SGD	Singapore	Singapore dollar
THB	Thailand	Thai baht
TJS	Tajikistan	Tajikistani somoni
TMT	Turkmenistan	Turkmenistan manat
TRY	Turkey	Turkish lira
UAH	Ukraine	Ukrainian hryvnia
USD	USA	United States dollar
UYU	Uruguay	Uruguayan peso

Currency code	Country	Currency name
VEF	Venezuela	Venezuelan bolívar
XAF	Communauté Financière Africaine (BEAC)	Central African CFA franc / CFA franc BEAC
ZAR	South Africa	South African rand

## Appendix 3. Language codes

Language code	Language
en	English
ru	Russian
de	German
tr	Turkish
nl	Dutch, Netherlands
pl	Polish
ro	Romanian
gr	Greek
fr	French
it	Italian
al	Albanian
bp	Brazilian Portuguese
bg	Bulgarian
ca	Catalan
cn, zh, zh-cn, zh-hans, zh-hant	Chinese
hr	Croatian

Language code	Language
cs	Czech
da	Danish
ee	Estonian
fi	Finnish
he	Hebrew
hu	Hungarian
ja	Japanese
no	Norwegian
pt	Portuguese
sr	Serbian
sk	Slovakian
sl	Slovenian
es	Spanish
sv	Swedish
th	Thai
la	Lao
uk	Ukrainian
ar	Arabic