**Seamless integration**

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Sept 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172128&selectedPageVersions=32&selectedPageVersions=33) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172128/www.atlassian.com)

**For easier integration, we have prepared for you:**

[**Tool for faster integration (Postman and Insomnia)**](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/2873163781/Tool+for+faster+integration)

**For all question you can check here:**[**20) Seamless interface F.A.Q.**](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.)

**If you don't get the question, you can write to us in the chat**

**Functions marked with \* are mandatory, other functions are optional**

**Seamless wallet**

Seamless wallet allows players to place bets directly to external system without requiring to transfer funds between accounts.

**Communication overview**

Communication between servers go through our secure HTTPS RESTful service. The service must be implemented by the external system according to specification.

The Game Engine sends GET requests to the remote system service and expects a JSON response with a status code and the players balance after the transaction

Example:

{"status":"200","balance":"300"}

**The status is not the same as HTTP status codes and only 200  (HTTP/1.1 - 200 OK) is accepted and allowed as a response to requests.**

**Request validation**

Each request is escorted with a query param key with a value of a sha1 cached query string and a salt key provided by Xapi Platform

Example:

action=balance&remote\_id=123&session\_id=123- abc&key=38432ff064690c9b03da519d0c685b1045451c9e).

The query param key value must match sha1([SALT KEY]+[QUERY STRING]

**Query string example:** action=balance&remote\_id=123&session\_id=123-abc

**Php code quick example**

|  |
| --- |
| $data = $\_GET;  $key = $data['key'];  unset($data['key']);  $hash = sha1($salt.http\_build\_query($data));      Then check if $key == $hash…. |

**Integration**

Before we start with wallet integration please provide us with two sets (staging and production) of the following:

* An IP list for us to whitelist
* An API endpoint URL where the wallet requests will be made (balance,debit,credit,rollback)

After we whitelist the addresses provided, we will provide you following details:

* APIi Endpoint url for Xapi calls (Create player, Reporting, etc.) - **Please ensure that endpoint is not setup behind CloudFlare**
* API credentials for our incoming calls (Xapi:Create player,reporting, etc.)
* API credentials for our outgoing calls to your wallet (balance,debit,credit,etc.) (Usualy the same credentials as incoming)

The [**dev.pantaloo.com**](https://dev.pantaloo.com/) domain in this document is not to be confused with the real domain. Please see email with api credentials for real development/production domain name.

External system must provide a secure https url that responds to the following GET requests with a JSON response with a status code and players balance after transaction.

You can use following rest testing form to start with integration:

 Expand source

We recommend you start with the simplest call [getGameList()](https://autaut.atlassian.net/wiki/pages/viewpage.action?pageId=90017398) in order to get started.

When you think that integration is ready for testing go to [Testing tool and Form](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172042/Testing+tool+and+Form)

If you have any questions regarding seamless wallet integration please take a look at [F.A.Q.](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172128/Seamless+integration) before contacting support.

| **Integration process:** | **In depth workflow of a integration** |
| --- | --- |
|  |  |

**20) Seamless interface F.A.Q.**

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Sept 29, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172096&selectedPageVersions=29&selectedPageVersions=30) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/www.atlassian.com)

* [INTEGRATION FLOW:](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-INTEGRATIONFLOW%3A)
* [Q: Details Basic flow](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ADetailsBasicflow)
* [FREEROUNDS](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-FREEROUNDS)
* [Q: How will we see requests coming from free spins (not freerounds)?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowwillweseerequestscomingfromfreespins(notfreerounds)%3F)
* [Q: How will we see requests coming from free rounds (not free spins)?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowwillweseerequestscomingfromfreerounds(notfreespins)%3F)
* [Q: How you can know if the request is Freerounds win?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowyoucanknowiftherequestisFreeroundswin%3F)
* [Q: Adding Freerounds for both version mobile and desktop](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AAddingFreeroundsforbothversionmobileanddesktop)
* [Q: min, mid, max values of each games that supports Freerounds](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3Amin%2Cmid%2CmaxvaluesofeachgamesthatsupportsFreerounds)
* [Q: RemoveFreerounds](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ARemoveFreerounds)
* [Q: Freerounds: validTo / validFrom](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AFreerounds%3AvalidTo%2FvalidFrom)
* [Q: Will we receive the gameid parameter on all requests?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWillwereceivethegameidparameteronallrequests%3F)
* [Q: In balance request, where will the gameid be sent, in which parameter?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AInbalancerequest%2Cwherewillthegameidbesent%2Cinwhichparameter%3F)
* [Q: You specify various data types, but we received only strings.](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AYouspecifyvariousdatatypes%2Cbutwereceivedonlystrings.)
* [Q: In the table with the json response, the transaction\_id is described as mandatory, but in the example of the credit call, it is not mentioned in the response, is it mandatory?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AInthetablewiththejsonresponse%2Cthetransaction_idisdescribedasmandatory%2Cbutintheexampleofthecreditcall%2Citisnotmentionedintheresponse%2Cisitmandatory%3F)
* [Q: In all credit/debit examples, the transaction\_id in the response does not match the transaction\_id in the request, should it match?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AInallcredit%2Fdebitexamples%2Cthetransaction_idintheresponsedoesnotmatchthetransaction_idintherequest%2Cshoulditmatch%3F)
* [Q: The transaction\_id's are described as being an integer, but all requests we received so far are prepended with 'gs-'. we assume that we can strip the non integer part and still receive unique id's, is this correct?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AThetransaction_id'saredescribedasbeinganinteger%2Cbutallrequestswereceivedsofarareprependedwith'gs-'.weassumethatwecanstripthenonintegerpartandstillreceiveuniqueid's%2Cisthiscorrect%3F)
* [Q: How should the response look like If we get the request with the transaction\_id that was already processedon our side?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowshouldtheresponselooklikeIfwegettherequestwiththetransaction_idthatwasalreadyprocessedonourside%3F)
* [Q: Can I successfully process the debit/credit request with the same 'transaction\_id'?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ACanIsuccessfullyprocessthedebit%2Fcreditrequestwiththesame'transaction_id'%3F)
* [Q: Why does some requests take so long to proccess?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhydoessomerequeststakesolongtoproccess%3F)
* [Q: How will we see if the player has won a jackpot?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowwillweseeiftheplayerhaswonajackpot%3F)
* [Q: Should the rollback be allowed even if 'round\_id' parameter is not present?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AShouldtherollbackbeallowedevenif'round_id'parameterisnotpresent%3F)
* [Q: Which amount value should we use in case of the rollback request? That one from the request or that from the transaction that is going to be rollbacked?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhichamountvalueshouldweuseincaseoftherollbackrequest%3FThatonefromtherequestorthatfromthetransactionthatisgoingtoberollbacked%3F)
* [Q: How should the response look like if we get the request(debit/credit) with the negative amount?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowshouldtheresponselooklikeifwegettherequest(debit%2Fcredit)withthenegativeamount%3F)
* [Q: Is 'roundId' unique per user per game? For example, if two players play roulette on same table at same time and place their bets, is their roundId different or same?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AIs'roundId'uniqueperuserpergame%3FForexample%2Ciftwoplayersplayrouletteonsametableatsametimeandplacetheirbets%2CistheirroundIddifferentorsame%3F)
* [Q: What are the differences between NE, NP and NL?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhatarethedifferencesbetweenNE%2CNPandNL%3F)
* [Q: My game is stuck on loading screen, it does not load further](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AMygameisstuckonloadingscreen%2Citdoesnotloadfurther)
* [Q: Microgaming games are not running on development environment.](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AMicrogaminggamesarenotrunningondevelopmentenvironment.)
* [Q: Authentication pops up on staging environment](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AAuthenticationpopsuponstagingenvironment)
* [Q: Calling images to frontend](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ACallingimagestofrontend)
* [Q: Which games can you test the integration on?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhichgamescanyoutesttheintegrationon%3F)
* [Q: rollback the transaction that not exists. ERROR 404 expected](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3Arollbackthetransactionthatnotexists.ERROR404expected)
* [Q: Slat Key test wrong in “basic tests”](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ASlatKeytestwrongin%E2%80%9Cbasictests%E2%80%9D)
* [Q: How can i use check key in my code?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowcaniusecheckkeyinmycode%3F)
* [Q: What is Gameplay\_final parameter?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhatisGameplay_finalparameter%3F)
* [Q: Show custom messages for returning response (insufficient funds)](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AShowcustommessagesforreturningresponse(insufficientfunds))
* [Q: Balance, Debit, Credit, Rollback](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ABalance%2CDebit%2CCredit%2CRollback)
* [Q: Main difference seamless and wallet](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AMaindifferenceseamlessandwallet)
* [Q: game\_id in the balance request isn't a mandatory parameter](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3Agame_idinthebalancerequestisn'tamandatoryparameter)
* [Q: If the transaction 1 happens now and 10 minutes lates we get duplicate transaction again, there is any chance that can happen?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AIfthetransaction1happensnowand10minuteslateswegetduplicatetransactionagain%2Cthereisanychancethatcanhappen%3F)
* [Q: Where I can test my integration?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhereIcantestmyintegration%3F)
* [Q: LoginPlayer is not mentioned in the workflow and when its called also the logoutPlayer](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ALoginPlayerisnotmentionedintheworkflowandwhenitscalledalsothelogoutPlayer)
* [Q: createPlayer: Do we get the same remote\_id for one username when we call the createPlayer or its always different?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AcreatePlayer%3ADowegetthesameremote_idforoneusernamewhenwecallthecreatePlayeroritsalwaysdifferent%3F)
* [Q: CreatePlayer: How can i actualize balance of a player stored on your server?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ACreatePlayer%3AHowcaniactualizebalanceofaplayerstoredonyourserver%3F)
* [Q: createPlayer: If you get error: “Payer already registered under a different agent« -> (CreatePlayer And PlayerExits)](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AcreatePlayer%3AIfyougeterror%3A%E2%80%9CPayeralreadyregisteredunderadifferentagent%C2%AB-%3E(CreatePlayerAndPlayerExits))
* [Q: When you need use GET or POST?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhenyouneeduseGETorPOST%3F)
* [Q: How can I know which methods are necessary? Do you have a list of most necessary methods?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowcanIknowwhichmethodsarenecessary%3FDoyouhavealistofmostnecessarymethods%3F)
* [Q: Rollback question: if not passed amount in rollback transaction, user balance will be same?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ARollbackquestion%3Aifnotpassedamountinrollbacktransaction%2Cuserbalancewillbesame%3F)
* [Q: Difference between getGame (play for fun) and getGameDemo](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ADifferencebetweengetGame(playforfun)andgetGameDemo)
* [Q: Which currency format you must use?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhichcurrencyformatyoumustuse%3F)
* [Q: Error fetching remote](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AErrorfetchingremote)
* [Q: How to to test the rollback?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowtototesttherollback%3F)
* [Q: Can we have freerounds and free spins in the same request?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ACanwehavefreeroundsandfreespinsinthesamerequest%3F)
* [Q: Storing transaction on their side](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AStoringtransactionontheirside)
* [Q: How to integrate jackpot?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowtointegratejackpot%3F)
* [Q: Possible bets level on your side?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3APossiblebetslevelonyourside%3F)
* [Q: Which crypto currencies we support?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhichcryptocurrencieswesupport%3F)
* [Q: Can integrate on stage Account providers, which not allow without license?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ACanintegrateonstageAccountproviders%2Cwhichnotallowwithoutlicense%3F)
* [Q: callback for production and stage can be the same?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3Acallbackforproductionandstagecanbethesame%3F)
* [Q: DIGITAIN](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ADIGITAIN)
* [Q: what is difference when you call getGameList with different currencies?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AwhatisdifferencewhenyoucallgetGameListwithdifferentcurrencies%3F)
* [Q: Rollback timeout. What is the usual flow if you get time out?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ARollbacktimeout.Whatistheusualflowifyougettimeout%3F)
* [Q: Is the issue of expiring session tokens after 24 hours?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AIstheissueofexpiringsessiontokensafter24hours%3F)
* [Q: How long should our session last?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowlongshouldoursessionlast%3F)
* [Q: what parts in your API we need to pay attention to so that we can distinguish between a regular bet and a free spin bet or, for example, a regular win from a jackpot win?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AwhatpartsinyourAPIweneedtopayattentiontosothatwecandistinguishbetweenaregularbetandafreespinbetor%2Cforexample%2Caregularwinfromajackpotwin%3F)
* [Q: Difference between "freespins" and "freerounds\_supported" parameters](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ADifferencebetween%22freespins%22and%22freerounds_supported%22parameters)
* [Q: if the games you are seeing in getGameList on STG are available or you need to enable/configure something additional from your side](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AifthegamesyouareseeingingetGameListonSTGareavailableoryouneedtoenable%2Fconfiguresomethingadditionalfromyourside)
* [Q: bonus\_buy: how can I filter games who support bonus by?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3Abonus_buy%3AhowcanIfiltergameswhosupportbonusby%3F)
* [Q: How to know which games support specified currency](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowtoknowwhichgamessupportspecifiedcurrency)
* [Q: Which one should you use (GetGame, getGameDirect, getGameDemo) ?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhichoneshouldyouuse(GetGame%2CgetGameDirect%2CgetGameDemo)%3F)
* [Q: How to get “megaways” games from getgamelist call](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowtoget%E2%80%9Cmegaways%E2%80%9Dgamesfromgetgamelistcall)
* [Q: cURL error 7: Failed to connect. Couldn't connect to server. Could the firewall be blocking it?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AcURLerror7%3AFailedtoconnect.Couldn'tconnecttoserver.Couldthefirewallbeblockingit%3F)
* [Q: how many currencies we have for games?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3Ahowmanycurrencieswehaveforgames%3F)
* [Q: Is user can have 2 currencies?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AIsusercanhave2currencies%3F)
* [Q: Possible ways to get hashing wrong](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3APossiblewaystogethashingwrong)
* [Q: Find all jackpot games](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AFindalljackpotgames)
* [Q: If you need jackpot type](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AIfyouneedjackpottype)
* [Q: How to use ip for whitlising if you have dynamics IPs?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowtouseipforwhitlisingifyouhavedynamicsIPs%3F)
* [Q: Is it possible to set multiple callbacks for multiple domains?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AIsitpossibletosetmultiplecallbacksformultipledomains%3F)
* [Q: Evolution restriction](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AEvolutionrestriction)
* [Q: Mobile and desktop version of the games](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AMobileanddesktopversionofthegames)
* [Q: What category is virtual-games?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhatcategoryisvirtual-games%3F)
* [Q: Why do you need to provide us with a callback URL when starting the integration?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhydoyouneedtoprovideuswithacallbackURLwhenstartingtheintegration%3F)
* [Q: documentation it says our endpoint cannot be behind cloudflare](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3Adocumentationitsaysourendpointcannotbebehindcloudflare)
* [Q: In documentation: jackpot\_contribution\_in\_amount is defined as double(10,6). And you cant parse false as double.](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AIndocumentation%3Ajackpot_contribution_in_amountisdefinedasdouble(10%2C6).Andyoucantparsefalseasdouble.)
* [Q: We send credit request with parameters amount:0 if users doesn’t win anything. What does transaction\_id represent them and if its possible to receive rollback with amount = 0?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWesendcreditrequestwithparametersamount%3A0ifusersdoesn%E2%80%99twinanything.Whatdoestransaction_idrepresentthemandifitspossibletoreceiverollbackwithamount%3D0%3F)
* [Q: Are sportsbook debit and credit requests the same as casino requests? Is there more data coming in?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AAresportsbookdebitandcreditrequeststhesameascasinorequests%3FIstheremoredatacomingin%3F)
* [Q: Is it mandatory to return balance field in every error response (status 404, 500, 400, ...)? What if you cant return balance in the moment of the error (for example query validation failed, internal server error happened)?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AIsitmandatorytoreturnbalancefieldineveryerrorresponse(status404%2C500%2C400%2C...)%3FWhatifyoucantreturnbalanceinthemomentoftheerror(forexamplequeryvalidationfailed%2Cinternalservererrorhappened)%3F)
* [Q: What should we do with a credit callbacks that have is\_promo\_win set to true? What are promo wins and when can we expect them?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AWhatshouldwedowithacreditcallbacksthathaveis_promo_winsettotrue%3FWhatarepromowinsandwhencanweexpectthem%3F)
* [Q: Difference between session\_id and gamesession\_id as both of them are sent in response of getGame.  Which one shall we verify for validity when balance is debited?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ADifferencebetweensession_idandgamesession_idasbothofthemaresentinresponseofgetGame.Whichoneshallweverifyforvaliditywhenbalanceisdebited%3F)
* [Q: How can we use our own sessionid and send it into getgame](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AHowcanweuseourownsessionidandsenditintogetgame)
* [Q: Is callerId always equal to api\_username and callerPassword always equal api\_password?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3AIscallerIdalwaysequaltoapi_usernameandcallerPasswordalwaysequalapi_password%3F)
* [Q: Do callerPassword and api\_password have the same value?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3ADocallerPasswordandapi_passwordhavethesamevalue%3F)
* [Q: why we send game\_id: null in balance request?](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172096/20+Seamless+interface+F.A.Q.#id-20)SeamlessinterfaceF.A.Q.-Q%3Awhywesendgame_id%3Anullinbalancerequest%3F)

**Integration tips**:

* please use game\_hash\_id for your game library, because it is the same between staging and production. game\_id will be different between staging and production
* transaction id is unique per player! two different players can share the same transaction id in some cases. Always take the remote\_id in consideration when processing transactions
* seamless requests from mobile game can have desktop game id

**INTEGRATION FLOW:**

**Q: Details Basic flow**

once a player clicks on a game he wants to play, you would need to send a playerExists request to our server, which will then either confirm or deny the players existance, if the player exists, cool, you send US a getGame request to OUR endpoint([stage.game-program.com/api/seamless/provider](http://stage.game-program.com/api/seamless/provider)). If the player does not yet exist, you must first createPlayer and then proceed with sending the getGame request for game launch(which will return the game url which you then insert into an iframe via src='[https://gameurl.com](https://gameurl.com/)..', and it will start loading the game.

We send a BALANCE request to YOUR endpoint(you need to create an endpoint that will accept such requests), you respond to the BALANCE request with a json format {"status":"200","balance":250} <-- in this case, 250 is the users balance, which you keep on your side. Balance requests will come after the game launch URL is used/clicked on and the game has started to load(in most cases).

The user makes a spin ingame for 5€, we will send a DEBIT(bet) request to YOUR endpoint(the one you need to set up), and you need to update the users balance accordingly(remove the bet amount from his balance) and respond with the updated balance as the response example above --> {"status":"200","balance":245}

After the debit request, a CREDIT(win) request will be sent to YOUR endpoint, and you need to update the users balance(if he didnt win anything, amount in request will be 0. If he wins 10€ anything, the amount will be set the win amount in the request), and respond with the updated users balance --> {"status":"200","balance":255}

Gameplay will cycle in this order, DEBIT request, CREDIT request, etc. and user can close the game whenever.

If any error happens while processing DEBIT or CREDIT requests, you will receive a ROLLBACK request, which you have to process(search for the DEBIT or CREDIT in your database, and return/take money from the user, as if the transaction has never happened).

**FREEROUNDS**

**Q: How will we see requests coming from free spins (not freerounds)?**

A: Order of events in free spins (same as normal bet / win)

1. you receive a bet request for 0.20 (for example)
2. then no win comes because player starts free spins (let them be 5 for example)
3. player plays all the free spins - during this time no communication will be between 'gameserver' and your wallet
4. when all free spins are played (finished) you receive a single win request with sum accumulated from all free spins winnings

So based on this scenario your wallet will see single bet 0.20 and single win 1.50 at the end and all free spins will be transparent for you.

**Q: How will we see requests coming from free rounds (not free spins)?**

Bet coming from free rounds will have is\_freeround\_bet = true and win will have is\_freeround\_win = true

A: When the player receives free rounds, this will be the order of events on your wallet:

1. you receive bet with amount 0, type of bet 'freeRoundBet'
2. when the player is finished using free rounds, you will either receive a positive or a 0 win of type 'freeRoundWin'.

**Q: How you can know if the request is Freerounds win?**

You have “is\_freeround\_win” so you would know it was winning from free rounds you awarded to player

**Q: Adding Freerounds for both version mobile and desktop**

If you add freer spins for both version (mobile and desktop) and launch the desktop version of the game first, you will be able to take advantage of the free spins there. But if you then open the mobile version, you wont have them there anymore. So if it’s the same game, the freespins will appear on the version you run first

**Q: min, mid, max values of each games that supports Freerounds**

Min -> Minimum bet or bet around 0,10€

Mid -> around 1€

Max -> from 3-5€

But this is depends from provider to provider

**Q: RemoveFreerounds**

When you create a freespins promotion you get freeround\_id under thath freeround\_id, there can be single game or multiple. But when you user RemoveFreerounds method, you delete that freeround promotion for that player (or multiple player)

So if you want to remove single game, you’d need to delte whole freeround and create it again

**Q: Freerounds: validTo / validFrom**

validFrom -> is a date field which contains. The date where the freerounds will be available from, in most cases it will be the same day the freerounds will be awarded. So, as an example, you will receive addFreeRounds request with validFrom set to 5.4.2022 and validTo set to 6.4.2022 for instance. You must treat these freerounds like they are available from 5.4.2022 0:00, until 6.4.2022 23:59.

**Q: Will we receive the gameid parameter on all requests?**

A: The gameid should always be present except on rollback. In some rare exceptions for example when player enters the livecasino lobby and gameid is unknown, we will send the livecasino lobby gameid.

**Q: In balance request, where will the gameid be sent, in which parameter?**

A: The values are sent in key⇒value combination, the order itself is not defined.

**Q: You specify various data types, but we received only strings.**

A: Please expect string values, but they should be treated as type described.

**Q: In the table with the json response, the transaction\_id is described as mandatory, but in the example of the credit call, it is not mentioned in the response, is it mandatory?**

A: It's not mandatory, but you can return it for future use. Use 'transactionid'.

**Q: In all credit/debit examples, the transaction\_id in the response does not match the transaction\_id in the request, should it match?**

A: That's just an example.

**Q: The transaction\_id's are described as being an integer, but all requests we received so far are prepended with 'gs-'. we assume that we can strip the non integer part and still receive unique id's, is this correct?**

A: Unfortunately 'gs - ' is a provider prefix, and its there to ensure that we send you unique ids because we have a lot of providers. But if you can use this prefix as a provider name, and use the numeric part as transaction id and still handle it as unique, than this is ok.

**Q: How should the response look like If we get the request with the transaction\_id that was already processedon our side?**

A: Example: If transaction\_id = 61385912731123 response was {“status”:“200”,“balance”:“105793.30”,“transactionId”:“362984”}, the response on the request with the same transaction\_id should be {“status”:“200”,“balance”:“105793.3000”,“transactionId”:“362984”} . Meaning that on a retry, the response of the transaction should be the same as on first request ..

**Q: Can I successfully process the debit/credit request with the same 'transaction\_id'?**

A: No! The possibility that you will get such a case is close to 0, but you'll never know. If you receive the same transaction id, please return same response as on first request (balance field should indicate current player balance)

**Q: Why does some requests take so long to proccess?**

A: We have a timeout simulation enabled on staging, so one in 10 spins may be slower up to 60 seconds so we can test the wallet properly

**Q: How will we see if the player has won a jackpot?**

A: You will receive a normal bet, and the jackpot win will be sent tied to the same roundid but it will be of type 'jackpotWin'. is\_jackpot\_win will be true.

**Q: Should the rollback be allowed even if 'round\_id' parameter is not present?**

A: Yes!

**Q: Which amount value should we use in case of the rollback request? That one from the request or that from the transaction that is going to be rollbacked?**

A: Second one. Use the amount from the transaction that is already stored on your side and not from the request.

**Q: How should the response look like if we get the request(debit/credit) with the negative amount?**

A: Please return something like {“status”:“500”,“msg”:“Negative amount not allowed!”}.

**Q: Is 'roundId' unique per user per game? For example, if two players play roulette on same table at same time and place their bets, is their roundId different or same?**

A: It can be same, or different, depending on provider honestly speaking. On slots it's always different, on livecasino it will be same. You shouldn't rely on it being the same across multiple players.

**Q: What are the differences between NE, NP and NL?**

A: For netent you have three types of games:

1. NE . normal netent games
2. NP - netent premium games (3% higher royalties)
3. NL - netent live casino

**Q: My game is stuck on loading screen, it does not load further**

A: Probably there's an error on your end, please check your logs and check your wallet implementation on your end. The game is making a balance request on your service and you're not returning the balance correctly

**Q: Microgaming games are not running on development environment.**

If Microgaming games are not working please add following lines to your host file:

41.223.121.106  WebServer8

41.223.121.106  [WebServer8.bluemesa.mgsops.net](http://webserver8.bluemesa.mgsops.net/)

41.223.121.105 WebServer4

41.223.121.105 [WebServer4.bluemesa.mgsops.net](http://webserver4.bluemesa.mgsops.net/)

**Q: Authentication pops up on staging environment**

On staging we are using additional AUTH with some providers. Use test / test to authenticate.

**Q: Calling images to frontend**

First you need to save all images, which you get from getGameList response into your database. And then you can show images on your website. You cant call directly from frontend to our API

**Q: Which games can you test the integration on?**

I recommend you try the integration in Habanero or GameArt games

**Q: rollback the transaction that not exists. ERROR 404 expected**

If the transaction is sent and an error occurs or the transaction fails, the same transaction is sent again. So in the second when the transaction is sent the status must be the same as in the first sent transaction

**Q: Slat Key test wrong in “basic tests”**

On this test with incorrect salt key, you can response status 403 and msg: “Hash Code Invalid” / “Wrong key”, etc… . The salt was given to you together with api username and password. In wallet requests you will find “key” parameter which is passed with each request. That key is generated with sha1 encoding and its build with salt and request query parameters. KEY = sha1([SALT KEY] + [QUERY STRING])

**Q: How can i use check key in my code?**

When you receive wallet requests for example balance, you first check that key parameter if is is correct continue… if its not correct return error

**Q: What is Gameplay\_final parameter?**

The "gameplay\_final" parameter in debit/credit requests will tell you if the round is finished or not. Debit requests will have gameplay\_final -> 0, credits gameplay\_final -> 1, meaning the round has finished.

**Q: Show custom messages for returning response (insufficient funds)**

We currently dont support any way of displaying custom messages on insufficient funds error for instance, as it depends from provider to provider, some support such things, some don’t.

**Q: Balance, Debit, Credit, Rollback**

**Balance**: requests users balance, you respond with status: 200, balance: 450.99 for instance

**Debit**: debit requests are called when user makes a bet, so its basically a bet request, which includes the amount used to lower player balance

requests users balance to be lowered(its a bet, so you deduce users balance by the bet amount in the request) and respond with status: 200, balance: 449.99 for instance

**Credit**: credit requests are called after the debit request, as a response/win. Includes amount to a credit to player

requests users balance to be added(its a win/end of round, so either the user wins something or doesnt) by the amount that is in the request, and you respond with status: 200, balance: 454.24

**Rollback**: Rollback are sent if any timeouts or other issues happened with debit/credit requests or response. For instance, we send you a debit request, you handle the debit request on your side and for some reasons we don't receive a response and the game crashes. A rollback request will be sent for the debit to player receives the balance back

**Q: Main difference seamless and wallet**

Essentially, with wallet you only transfer balance before player goes in game from your agent to player, while with seamless you require database not just for balances but also for transactions, so you have to store all individual transactions and operate them - modify player balance based on the requests you get from us.

**wallet** is actually easier and more secure - considering that it is our game server directly in contact with the game provider.

**seamless** all balance is updated in real time, as we are communicating with you to update balance.

**Q: game\_id in the balance request isn't a mandatory parameter**

place for game\_id on balance requests is therefore incorrect as the request varies from provider to provider

**Q: If the transaction 1 happens now and 10 minutes lates we get duplicate transaction again, there is any chance that can happen?**

That could happen, I don’t believe its happening regulary with basic transaction bit it. But usually dupliacate transaction happen very quickly one after another as part of a retry

**Q: Where I can test my integration?**

We have tests online from the backoffice, which you can run for yourself

Instructions on the tests are available here (under entry - Testing form):

<https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172042/Testing+tool+and+Form>

**Once you pass all the tests, i will have to retest and check that all the tests are successful, before the project moves to production**

**Q: LoginPlayer is not mentioned in the workflow and when its called also the logoutPlayer**

Generally it is not needed, loginPlayer and logoutPlayer are used to set and unset playing sessions for players and can also be used in that regard to check if session is still active and reject whatever gameplay comes from unactive session, but we also gave it handled on our side

**Q: createPlayer: Do we get the same remote\_id for one username when we call the createPlayer or its always different?**

You will receive a unique remote\_id for a player once you create him using createPlayer method. If you calling that again for the same user you will receive a “Player already exists” error, so instead, you can call playerExists, which will return user data. I would  advise to do it like this: when user click on a game, call playerExists if you receive a response that the player doesn’t exist yet, call createPlayer, and

**Q: CreatePlayer: How can i actualize balance of a player stored on your server?**

Balance of a player is completely handled on your side, meaning you will have to take care of adding or reducing players balance. We only store player data like id, username, and some other meaningless info, while you have to keep track of the ids, usernames, balances, etc.

**Q: createPlayer: If you get error: “Payer already registered under a different agent« -> (CreatePlayer And PlayerExits)**

previously created player with same username but under different currency. Player usernames are unique on our side.

If the user "test123" is created under the EUR, then you must set the currency parameter to the value that this user belongs to (in this case on EUR)

**Q: When you need use GET or POST?**

**GET** -> for balance , credit, debit, callback

**POST** -> for all other methods (getGame, createPlayer, etc.)

**Q: How can I know which methods are necessary? Do you have a list of most necessary methods?**

Functions marked with \* are mandatory, other functions are optional

**Q: Rollback question: if not passed amount in rollback transaction, user balance will be same?**

No, you should keep track of transactions and transaction id’s on your side and when a refund/rollback request is received by you, you should find transaction in your database using only the transaction id from the rollback request, and return the amount to/from the player. But rely only on transaction id in the rollback request, take the amount and everything else from your side when you locate transaction

**Q: Difference between getGame (play for fun) and getGameDemo**

**GetGame** always return URL you should load directly. Is used for normal games: slots/livecasino/virtual etc. You can get demo version of slots/livecasino/virtual games by passing an additional parameter play\_for\_fun = 1 in getGame request. In play for fun mode player can spin and play slot, but he is not playing with real money. Here you have free credits. It is used mostly for slot games.

**GetGameDemo** is used for demo version of sportsbook. In this case player cannot place any bet or play game. He can just open and see how game looks like. It is used for sportbetting to open sportbetting and see what option he has to bet. Normally player can only select ticket in demo mode he would like to play, but he cannot place a bet until he does login as real player and only then he can place bet with his real balance.

**Q: Which currency format you must use?**

We use the standard EUR and USD… three character format

**Q: Error fetching remote**

Error fetching remote means that your endpoint is unreachable, meaning you'll have to take a look at the issue with your system administrators as its something related to the servers.

**Q: How to to test the rollback?**

Place a 10 second sleep in the code before responding to debit requests, and try to place a bet

So you'll answer the debit request after 10 seconds, which should trigger a rollback

Also, please test this using Habanero provider, as it depends from provider to provider how everything is handled after a failed debit, Habanero should respond with a rollback if the debit is answered after 10 seconds

**Q: Can we have freerounds and free spins in the same request?**

Depends from provider to provider . Some games providers send credit eaxh spin(also if winning is 0), others send only if there was win in spin (e.g. from 10 awarded spin 3 are winning – you will get 3 requests). Most of them send credit request only at the end (at last spin). So, you cannot rely on counting freespins

**Q: Storing transaction on their side**

You should store each transaction on your side anyway, you will need transaction data to resolve issues when players complain to you.

**Q: How to integrate jackpot?**

We don’t gave any such functionality implemented on out side, we suggest you fetch jackpot values once per day as incrementing based on no ingo isn’t accurate at all

**Q: Possible bets level on your side?**

Bet levels depends on provider. We do not have a list of possible bets

**Q: Which crypto currencies we support?**

Bitcoin, Mili Bitcoin (MBC), Micro Bitcoin (UBC), Mili BTC Cash(MBH), Ethereum, Litecoin

**Q: Can integrate on stage Account providers, which not allow without license?**

Usually if you don’t have provider in the contract, we don’t enable them for you on staging, as you wont need them, so  no, if you don’t gave license we wont enable them

**Q: callback for production and stage can be the same?**

I strongly suggest having different callbacks for production and staging environment, as if any mixup happens, it could be with real money or vice versa, or you could accidentally test something on production instead of staging etc.

**Q: DIGITAIN**

* **Digitain how to launch the game:** please use getGameDirect method to call the game with its own game id, and you will receive the full embed code as a response, which contains code with already filled parameters.
* **CNAME has to be added on your side, where the site is hosted**
* run through the JS code in documentation, it is not possible to launch with valid token
* **What is CNAME?** -> Your website name

**Q: what is difference when you call getGameList with different currencies?**

the currency parameter is used for filtering games which support it or not, so if a game/provider doesnt support for instance AED currency, it will not show up in the getGameList response with currency: AED used

**Q: Rollback timeout. What is the usual flow if you get time out?**

Depends from provider to provider, but usually it goes like this: provider sends a bet, you fail to respond correctly to the bet the provider sends a rollback request and you rollbaxk the bet. So, if request times out its counted as invalid/not processed so you should receive the rollback request, if the transaction was processed on your side you can handle the rollback request normally. If not you should not handle and just respond with status 404.

**Q: Is the issue of expiring session tokens after 24 hours?**

Yes that will be the problem, because provider can keep retrying transaction for more than 24hr in scenario where you do not accept transaction from provider and you expire token the transaction can be lost

**Q: How long should our session last?**

The session must not be closed. Especially when it comes to winning. And you must return 200 and balance on credit an debit

**Q: what parts in your API we need to pay attention to so that we can distinguish between a regular bet and a free spin bet or, for example, a regular win from a jackpot win?**

There is no difference between a regular bet and in-game free spins bet.

In the case of a jackpot, the jp cpntribution parameter is sent in the bet request. If the jackpot is a win, the value of is\_jacpot\_win request parameter is set to 1 and jackpot\_win\_amount->amount of the win

**Q: Difference between "freespins" and "freerounds\_supported" parameters**

which give him 10 freespins. normally those pay out more than normal spin as it has some special feature

freerounds are what you award to player as "bet for free" ... so you give him 10 spins which are normal spins (not part of bonus game), just that bets for those spins are not deducted from player balance

**Q: if the games you are seeing in getGameList on STG are available or you need to enable/configure something additional from your side**

All the games you receive getGameList, are aviable for your API

**Q: bonus\_buy: how can I filter games who support bonus by?**

We are sending bonusbuy: true / false on getGameList request, but you have add “show\_additional” : true to the request to receive additional details, one of them being this bonus buy parameter. I believe not all games have this in info though

**Q: How to know which games support specified currency**

If you retrieve getGameList for USD currency, you will receive a response that contains games that support USD currency, if you call it for EUR currency, you will receive games that support EUR currency

**Q: Which one should you use (GetGame, getGameDirect, getGameDemo) ?**

GetGame -> is used for normal slot games, virtual games, live casinos etc.

getGameDirect -> is used for sportsbook providers as it returnes embed code

getGameDemo Is used for demo version of sportsbook

**Q: How to get “megaways” games from getgamelist call**

You can get games that are "megaways" with our getgamelist method. Add a parameter to the method call:

   "show\_additional" : true. Here's how to do it:

   "api\_password" : "api password",

    "api\_login" : api name

    "method" : "getGameList"

    "show\_systems" : 0,

    "show\_additional" : true,

    "currency" : "EUR"

**Q: cURL error 7: Failed to connect. Couldn't connect to server. Could the firewall be blocking it?**

It is possible that a firewall is blocking the connection to the server “[em-api.thegameprovider.com](http://em-api.thegameprovider.com/)” on port 443.

A firewall is a security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It can be implemented in hardware or software and can be set up to block specific IP addresses, ports or types of traffic.

If the server you are trying to connect to is located on a network that is behind a firewall, the connection may be blocked if the firewall is not configured to allow traffic on port 443.

It is recommended to check the firewall settings on your network or contact your network administrator to verify if the traffic to the server is being blocked.

If you are trying to connect to a server hosted by a third-party, you should contact their support team to check if the server is properly configured to accept your incoming connection.

**Q: how many currencies we have for games?**

we support any ISO currency + some popular crypto but then is up to providers if they support certain currency or not

**Q: Is user can have 2 currencies?**

No, you can only have one username in one currency. If you want to have the same username for multiple currencies, I suggest adding a prefix or something similar to the username

**Q: Possible ways to get hashing wrong**

We are sending jackpot\_contribution\_in\_amount = 0 via the GET request. Check request if you see jackpot\_contribution\_in\_amount = false. If yes then you need to handle parameter to 0.

**Q: Find all jackpot games**

To find out all the games that are jackpot, use the getGameList method. In the response you have parameter “has\_jackpot”. This parameter have two values: true or false

**Q: If you need jackpot type**

if you need the jackpot type, add the show\_additional: true parameter to the getGameList request. In the response you see a array of parameters.

**Q: How to use ip for whitlising if you have dynamics IPs?**

If you want develop locally and want to use dynamic IP, you can do, but you will have to update it by yourself in the backoffice. This is not possible on production

**Q: Is it possible to set multiple callbacks for multiple domains?**

If you want separate accesses for two separate website, you would have to request a new access so we can give you a new one that has a different endpoint set up, and you would use the new credential for game launches

**Q: Evolution restriction**

Evolution just implemented a currency restriction. This means that you can play with the currency that matchers the geo location from where you are launchink the game (e.g. TRY -turkey, INS-israel) or with any currency tthat stronger/more worth. Generic currencies like EUR, USD, EGP are excluded from this restrictions

**Q: Mobile and desktop version of the games**

We want to separate mobile and desktop games on out end to keep track on where the games was player

**Q: What category is virtual-games?**

Games are usually put into a separate category from live casinos and slots

**Q: Why do you need to provide us with a callback URL when starting the integration?**

You need to provide callback URL to which we will send waller requests when player start the game and play it. For example your callback url will be: <https://test.com/api> and we will be sending wallet requests on that url

**Q: documentation it says our endpoint cannot be behind cloudflare**

If you have no way of exposing URL from outside of Cloudflare infrastructure that is not a blocker, however we did have myriad of issues with clients who are having their API endpoint on Cloudflare.

* Potential errors that you may experience due to cloudflare?

1. We send 100 requests to you in a second, Cloudflare will mark us DDOS
2. If you are on Cloudflare, your ip will be changing and routes through which we are sending you requests will change too, which from experience bought bad experience to players, theirs spins taking long time or missing winning. Because CF decided some route is closer to us any point and debugging that issues in nearly impossible

**Q: In documentation: jackpot\_contribution\_in\_amount is defined as double(10,6). And you cant parse false as double.**

Its actually sent as 0, but while long parsing logs they take a form Boolean value for some reason, its actually a 0

**Q: We send credit request with parameters amount:0 if users doesn’t win anything. What does transaction\_id represent them and if its possible to receive rollback with amount = 0?**

Each bet/win is logged as separate transaction which has unique transaction\_id. That transaction\_id is also used when rollback request is sent, to know which transaction needs to be refunded

**Q: Are sportsbook debit and credit requests the same as casino requests? Is there more data coming in?**

the requests are the same yes, as for the payload, its the same, except we can add another parameter in the requests which includes additional bet and win data if you'd like. We more or less forward the whole request that the sportsbook provider sent us to you, so you get the whole data we get

**Q: Is it mandatory to return balance field in every error response (status 404, 500, 400, ...)? What if you cant return balance in the moment of the error (for example query validation failed, internal server error happened)?**

Status and balance are only mandatory in successful response cases, if the request fails for some reason you can respond with status and message, you can throw in the balance anyway and if its not possible to return it set it as null, doesnt really matter

**Q: What should we do with a credit callbacks that have is\_promo\_win set to true? What are promo wins and when can we expect them?**

Thats only signalling that the win is a promo win, generally promo wins come after the round has already completed, or even as a standalone round. Providers can have a promotion in one of their games, and if one of your players wins that promo, they can recieve a random "win" for some amount in that game. So for less confusion, its flagged

**Q: Difference between session\_id and gamesession\_id as both of them are sent in response of getGame.  Which one shall we verify for validity when balance is debited?**

"gamesession\_id" is used to track game sessions that is unique per game launch. So when you close current game, and open another, you will receive new "gamesession\_id". "sessionid" is used to track player login session. When for example player logins in your casino. You can use your own "sessionid" by sending "sessionid" parameter in getGame() request.

**Q: How can we use our own sessionid and send it into getgame**

You can use this parameter like others you send in getGame reuqest:

{

"api\_login" : "testapi",

"api\_password": "testapi",

"method": "getGame",

"lang" : "en",

"user\_username":"player1",

"user\_password":"player1",

"gameid" : "114575",

"sessionid": "12345678910",

"homeurl" : "[http://url.to.your.page.com](http://url.to.your.page.com/)",

"cashierurl" : "[http://url.to.cashier.page.com](http://url.to.cashier.page.com/)",

"play\_for\_fun" : 0,

"currency": "EUR"

}

You can use it if you need it. It is not mandatory parameter to send it in request

**Q: Is callerId always equal to api\_username and callerPassword always equal api\_password?**

API username is the callerID, and API password is callerPassword. Its the same think just different naming. You need to use your API credentials.

**Q: Do callerPassword and api\_password have the same value?**

yes, but callerPassword is sha1 encryption

**Q: why we send game\_id: null in balance request?**

Some providers do not send us the game ID. Therefore we cannot forward it. Game\_id is not mandatory in a balance request

**Tool for faster integration**

* 

Owned by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)

Last updated: [Mar 06, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=2873163781&selectedPageVersions=8&selectedPageVersions=9)

2 min read

We have created a gadget that will allow you to integrate more easily and faster. The documents are easy to use. They are made for the insomnia and postman program.

Below are two documents (Postman and Insomnia).

**INSOMNIA**

Open Insomnia integration

Insomnia integration

06 May 2022, 05:39 PM

**POSTMAN**

Open Postman Integration.json

Postman Integration.json

09 Aug 2022, 01:41 PM  
  
  
POSTMAN ::  
  
{

"version": 1,

"collections": [

{

"id": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"uid": "0-eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"name": "IntegrationStage",

"description": **null**,

"auth": **null**,

"events": **null**,

"variables": [

{

"key": "nejcd3\_mc\_s",

"value": "api\_password",

"disabled": false

}

],

"order": [

"95d820f1-1769-4803-b501-f49927b588df",

"86ccd196-0520-48cb-9601-dca8043c1ad9",

"60c3d212-938b-4c8d-8898-98bc60c560c7",

"f35ac06f-7081-477f-97b8-b28d83ace481",

"be662fa4-1243-496d-88ca-71087d2b2260",

"777adaf3-4449-4a29-a679-099a9259ca3d",

"7719be14-091b-4b1f-b150-95f8038f23cb",

"0a8e5274-3bce-40b8-829c-06f85b0aa1ed",

"6de9645c-0dce-4e01-8406-68cec55f24a6",

"926d5607-19d8-445f-bd87-97d2e93364b4",

"1f3bcd89-4816-41bf-b420-3186b1be80e8",

"d39beff6-74f2-4219-836c-148f0043d00c",

"84c6d8c5-5d44-4b84-bd96-7d01f2ed938d"

],

"folders\_order": [],

"protocolProfileBehavior": {},

"createdAt": "2022-05-04T11:18:29.000Z",

"folders": [],

"requests": [

{

"id": "0a8e5274-3bce-40b8-829c-06f85b0aa1ed",

"uid": "0-0a8e5274-3bce-40b8-829c-06f85b0aa1ed",

"name": "removeFreerounds",

"url": "{{urlstage}}",

"description": **null**,

"data": [],

"dataOptions": {

"raw": {

"language": "json"

}

},

"dataMode": "raw",

"headerData": [],

"method": "POST",

"pathVariableData": [],

"queryParams": [],

"auth": **null**,

"events": **null**,

"folder": **null**,

"responses\_order": [],

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"rawModeData": "{\r\n\t\"api\_login\" : \"{{api\_username}}\",\r\n \"api\_password\": \"{{api\_password}}\",\r\n\t\"method\": \"removeFreeRounds\",\r\n \"playerids\":\"\",\r\n\t\"gameids\":\"10497\",\r\n\t\"freeround\_id\": \"\",\r\n\t\"currency\": \"{{currency}}\"\r\n}",

"headers": "",

"pathVariables": {}

},

{

"id": "1f3bcd89-4816-41bf-b420-3186b1be80e8",

"uid": "0-1f3bcd89-4816-41bf-b420-3186b1be80e8",

"name": "debit",

"url": "{{urlstage}}?callerId={{api\_username}}&callerPassword={{api\_password}}&callerPrefix=700ha&action=debit&remote\_id&username=\"remote ID (player ID), NOT local (player ID)\"&session\_id=61152976d519b &amount=3&provider=ha&game\_id=10499&game\_id\_hash=ha\_ha-bird-of-thunder&transaction\_id=ha-10bc85c311fceb11b5630050f238e02b&round\_id=10bc85c3-11fc-eb11-b563-0050f238e02b&gameplay\_final=0&is\_freeround\_bet=0&jackpot\_contribution\_in\_amount=0&gamesession\_id=ha\_976201-10499-7be01bbb9be8786f4cc1cd55bd8c8a7e&=",

"description": **null**,

"data": **null**,

"dataOptions": **null**,

"dataMode": **null**,

"headerData": [],

"method": "GET",

"pathVariableData": [],

"queryParams": [

{

"key": "callerId",

"value": "{{api\_username}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "callerPassword",

"value": "{{api\_password}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "callerPrefix",

"value": "700ha",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "action",

"value": "debit",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "remote\_id",

"value": **null**,

"equals": false,

"description": **null**,

"enabled": true

},

{

"key": "username",

"value": "\"remote ID (player ID), NOT local (player ID)\"",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "session\_id",

"value": "61152976d519b ",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "amount",

"value": "3",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "provider",

"value": "ha",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "game\_id",

"value": "10499",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "game\_id\_hash",

"value": "ha\_ha-bird-of-thunder",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "transaction\_id",

"value": "ha-10bc85c311fceb11b5630050f238e02b",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "round\_id",

"value": "10bc85c3-11fc-eb11-b563-0050f238e02b",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "gameplay\_final",

"value": "0",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "is\_freeround\_bet",

"value": "0",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "jackpot\_contribution\_in\_amount",

"value": "0",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "gamesession\_id",

"value": "ha\_976201-10499-7be01bbb9be8786f4cc1cd55bd8c8a7e",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": **null**,

"value": "",

"equals": true,

"description": **null**,

"enabled": true

}

],

"auth": **null**,

"events": [

{

"listen": "prerequest",

"script": {

"id": "cd37b4d9-0a2b-43e3-8592-84b9d2e13e46",

"exec": [

"var str\_2 = pm.environment.get['api\_password'];\r",

"// Use the CryptoJS\r",

"var mdpassword = CryptoJS.MD5(str\_2).toString();\r",

"// Set the new environment variable\r",

"var a = postman.setEnvironmentVariable('api\_password', mdpassword);\r",

"\r",

"console.log(a);"

],

"type": "text/javascript"

}

}

],

"folder": **null**,

"responses\_order": [],

"preRequestScript": "var str\_2 = pm.environment.get['api\_password'];\r\n// Use the CryptoJS\r\nvar mdpassword = CryptoJS.MD5(str\_2).toString();\r\n// Set the new environment variable\r\nvar a = postman.setEnvironmentVariable('api\_password', mdpassword);\r\n\r\nconsole.log(a);",

"tests": **null**,

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"headers": "",

"pathVariables": {}

},

{

"id": "60c3d212-938b-4c8d-8898-98bc60c560c7",

"uid": "0-60c3d212-938b-4c8d-8898-98bc60c560c7",

"name": "playerExists",

"url": "{{urlstage}}",

"description": **null**,

"data": [],

"dataOptions": {

"raw": {

"language": "json"

}

},

"dataMode": "raw",

"headerData": [],

"method": "POST",

"pathVariableData": [],

"queryParams": [],

"auth": **null**,

"events": **null**,

"folder": **null**,

"responses\_order": [],

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"rawModeData": "{\r\n\t\"api\_password\" : {{api\_password}},\r\n\t\"api\_login\": {{api\_username}},\r\n\t\"method\": \"playerExists\",\r\n\t\"user\_username\": {{user\_username}},\r\n\t\"currency\": {{currency}}\r\n}",

"headers": "",

"pathVariables": {}

},

{

"id": "6de9645c-0dce-4e01-8406-68cec55f24a6",

"uid": "0-6de9645c-0dce-4e01-8406-68cec55f24a6",

"name": "getRoundHistory",

"url": "{{urlstage}}",

"description": **null**,

"data": [],

"dataOptions": {

"raw": {

"language": "json"

}

},

"dataMode": "raw",

"headerData": [],

"method": "POST",

"pathVariableData": [],

"queryParams": [],

"auth": **null**,

"events": **null**,

"folder": **null**,

"responses\_order": [],

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"rawModeData": "{\r\n\t\"api\_login\" : \"{{api\_username}}\",\r\n \"api\_password\": \"{{api\_password}}\",\r\n\t\"method\": \"getRoundHistory\",\r\n\t\"game\_id\" : 10497,\r\n\t\"round\_id\" : \"\",\r\n \"playerids\":\"\",\r\n\t\"currency\": \"{{currency}}\"\r\n}",

"headers": "",

"pathVariables": {}

},

{

"id": "7719be14-091b-4b1f-b150-95f8038f23cb",

"uid": "0-7719be14-091b-4b1f-b150-95f8038f23cb",

"name": "addFreerounds",

"url": "{{urlstage}}",

"description": **null**,

"data": [],

"dataOptions": {

"raw": {

"language": "json"

}

},

"dataMode": "raw",

"headerData": [],

"method": "POST",

"pathVariableData": [],

"queryParams": [],

"auth": **null**,

"events": **null**,

"folder": **null**,

"responses\_order": [],

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"rawModeData": "{\r\n\t\"api\_login\" : \"{{api\_username}}\",\r\n \"api\_password\": \"{{api\_password}}\",\r\n\t\"method\": \"addFreeRounds\",\r\n\t\"tittle\":\"TestAddFreeround\",\r\n \"playerids\":\"\",\r\n\t\"gameids\":\"10497\",\r\n\t\"available\": 2,\r\n\t\"validTo\":\"2022-01-12\",\r\n\t\t\"validFrom\":\"2022-01-12\",\r\n\t\t\"betlevel\" :\"min\",\r\n\t\t\"currency\": \"{{currency}}\"\r\n}",

"headers": "",

"pathVariables": {}

},

{

"id": "777adaf3-4449-4a29-a679-099a9259ca3d",

"uid": "0-777adaf3-4449-4a29-a679-099a9259ca3d",

"name": "getGameDirect",

"url": "{{urlstage}}",

"description": **null**,

"data": [],

"dataOptions": {

"raw": {

"language": "json"

}

},

"dataMode": "raw",

"headerData": [],

"method": "POST",

"pathVariableData": [],

"queryParams": [],

"auth": **null**,

"events": **null**,

"folder": **null**,

"responses\_order": [],

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"rawModeData": "{\r\n\t\"api\_login\" : {{api\_username}},\r\n \"api\_password\": \"{{api\_password}}\",\r\n\t\"method\": \"getGameDirect\",\r\n\t\"lang\" : \"{{lang}}\",\r\n\t\"user\_username\":\"{{user\_username}}\",\r\n\t\"user\_password\":\"{{user\_password}}\",\r\n\t\"gameid\" : 10497,\r\n\t\"homeurl\" : \"http://url.to.your.page.com\",\r\n\t\"play\_for\_fun\" : 0,\r\n\t\"currency\": \"{{currency}}\"\r\n}",

"headers": "",

"pathVariables": {}

},

{

"id": "84c6d8c5-5d44-4b84-bd96-7d01f2ed938d",

"uid": "0-84c6d8c5-5d44-4b84-bd96-7d01f2ed938d",

"name": "rollback",

"url": "{{urlstage}}?callerId={{api\_username}}&callerPassword={{api\_password}}&callerPrefix=700ha&action=credit&remote\_id=\"remote ID (player ID), NOT local (player ID)\"&username={{user\_username}}&session\_id=61152976d519b &amount=10&provider=ha&game\_id=10499&game\_id\_hash=ha\_ha-bird-of-thunder&transaction\_id=ha-10bc85c311fceb11b5630050f238e02b&round\_id=10bc85c3-11fc-eb11-b563-0050f238e02b&gameplay\_final=0&is\_freeround\_bet=0&jackpot\_contribution\_in\_amount=0&gamesession\_id=ha\_976201-10499-7be01bbb9be8786f4cc1cd55bd8c8a7e",

"description": **null**,

"data": **null**,

"dataOptions": **null**,

"dataMode": **null**,

"headerData": [],

"method": "GET",

"pathVariableData": [],

"queryParams": [

{

"key": "callerId",

"value": "{{api\_username}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "callerPassword",

"value": "{{api\_password}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "callerPrefix",

"value": "700ha",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "action",

"value": "credit",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "remote\_id",

"value": "\"remote ID (player ID), NOT local (player ID)\"",

"equals": true,

"description": "",

"enabled": true

},

{

"key": "username",

"value": "{{user\_username}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "session\_id",

"value": "61152976d519b ",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "amount",

"value": "10",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "provider",

"value": "ha",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "game\_id",

"value": "10499",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "game\_id\_hash",

"value": "ha\_ha-bird-of-thunder",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "transaction\_id",

"value": "ha-10bc85c311fceb11b5630050f238e02b",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "round\_id",

"value": "10bc85c3-11fc-eb11-b563-0050f238e02b",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "gameplay\_final",

"value": "0",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "is\_freeround\_bet",

"value": "0",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "jackpot\_contribution\_in\_amount",

"value": "0",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "gamesession\_id",

"value": "ha\_976201-10499-7be01bbb9be8786f4cc1cd55bd8c8a7e",

"equals": true,

"description": **null**,

"enabled": true

}

],

"auth": **null**,

"events": [

{

"listen": "prerequest",

"script": {

"id": "4e545f38-ef8b-4b60-a396-6995c3b2cb29",

"exec": [

"var str\_2 = pm.environment.get['api\_password'];\r",

"// Use the CryptoJS\r",

"var mdpassword = CryptoJS.MD5(str\_2).toString();\r",

"// Set the new environment variable\r",

"var a = postman.setEnvironmentVariable('api\_password', mdpassword);\r",

"\r",

"console.log(a);"

],

"type": "text/javascript"

}

}

],

"folder": **null**,

"responses\_order": [],

"preRequestScript": "var str\_2 = pm.environment.get['api\_password'];\r\n// Use the CryptoJS\r\nvar mdpassword = CryptoJS.MD5(str\_2).toString();\r\n// Set the new environment variable\r\nvar a = postman.setEnvironmentVariable('api\_password', mdpassword);\r\n\r\nconsole.log(a);",

"tests": **null**,

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"headers": "",

"pathVariables": {}

},

{

"id": "86ccd196-0520-48cb-9601-dca8043c1ad9",

"uid": "0-86ccd196-0520-48cb-9601-dca8043c1ad9",

"name": "createPlayer",

"url": "https://stage.game-program.com/api/seamless/provider",

"description": **null**,

"data": [],

"dataOptions": {

"raw": {

"language": "json"

}

},

"dataMode": "raw",

"headerData": [],

"method": "POST",

"pathVariableData": [],

"queryParams": [],

"auth": **null**,

"events": **null**,

"folder": **null**,

"responses\_order": [],

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"rawModeData": "{\r\n\t\"api\_password\" : {{api\_password}},\r\n\t\"api\_login\": {{api\_username}},\r\n\t\"method\": \"createPlayer\",\r\n\t\"user\_username\": {{user\_username}},\r\n\t\"user\_password\": {{user\_password}},\r\n\t\"user\_nickname\": \"\",\r\n\t\"currency\": {{currency}}\r\n}\r\n\r\n",

"headers": "",

"pathVariables": {}

},

{

"id": "926d5607-19d8-445f-bd87-97d2e93364b4",

"uid": "0-926d5607-19d8-445f-bd87-97d2e93364b4",

"name": "balance",

"url": "{{urlstage}}?callerId={{api\_username}}&callerPassword={{api\_password}}&callerPrefix=700h&action=balance&remote\_id=\"remote ID (player\_ID), NOT local (player\_ID)\"&username={{user\_username}}&session\_id=61152976d519b&currency={{currency}}&provider=ha&game\_id\_hash=ha\_ha-bird-of-thunder&key=sha1([SALT KEY]+[QUERY STRING]",

"description": **null**,

"data": **null**,

"dataOptions": **null**,

"dataMode": **null**,

"headerData": [],

"method": "GET",

"pathVariableData": [],

"queryParams": [

{

"key": "callerId",

"value": "{{api\_username}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "callerPassword",

"value": "{{api\_password}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "callerPrefix",

"value": "700h",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "action",

"value": "balance",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "remote\_id",

"value": "\"remote ID (player\_ID), NOT local (player\_ID)\"",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "username",

"value": "{{user\_username}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "session\_id",

"value": "61152976d519b",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "currency",

"value": "{{currency}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "provider",

"value": "ha",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "game\_id\_hash",

"value": "ha\_ha-bird-of-thunder",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "key",

"value": "sha1([SALT KEY]+[QUERY STRING]",

"equals": true,

"description": **null**,

"enabled": true

}

],

"auth": **null**,

"events": [

{

"listen": "prerequest",

"script": {

"id": "bbe0a56e-ac3b-4902-b423-dc3a210c2603",

"exec": [

"var str\_2 = pm.environment.get['api\_password'];\r",

"// Use the CryptoJS\r",

"var mdpassword = CryptoJS.MD5(str\_2).toString();\r",

"// Set the new environment variable\r",

"var a = postman.setEnvironmentVariable('api\_password', mdpassword);\r",

"\r",

"console.log(a);"

],

"type": "text/javascript"

}

}

],

"folder": **null**,

"responses\_order": [],

"preRequestScript": "var str\_2 = pm.environment.get['api\_password'];\r\n// Use the CryptoJS\r\nvar mdpassword = CryptoJS.MD5(str\_2).toString();\r\n// Set the new environment variable\r\nvar a = postman.setEnvironmentVariable('api\_password', mdpassword);\r\n\r\nconsole.log(a);",

"tests": **null**,

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"headers": "",

"pathVariables": {}

},

{

"id": "95d820f1-1769-4803-b501-f49927b588df",

"uid": "0-95d820f1-1769-4803-b501-f49927b588df",

"name": "getGameList",

"url": "https://stage.game-program.com/api/seamless/provider",

"description": **null**,

"data": [],

"dataOptions": {

"raw": {

"language": "json"

}

},

"dataMode": "raw",

"headerData": [],

"method": "POST",

"pathVariableData": [],

"queryParams": [],

"auth": **null**,

"events": **null**,

"folder": **null**,

"responses\_order": [],

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"rawModeData": "{\r\n\t\"api\_password\" : {{api\_password}},\r\n\t\"api\_login\": {{api\_username}},\r\n\t\"method\": \"getGameList\",\r\n\t\"show\_additional\": true,\r\n\t\"show\_systems\" : 1,\r\n\t\"currency\": {{currency}}\r\n}",

"headers": "",

"pathVariables": {}

},

{

"id": "be662fa4-1243-496d-88ca-71087d2b2260",

"uid": "0-be662fa4-1243-496d-88ca-71087d2b2260",

"name": "getGameDemo",

"url": "{{urlstage}}",

"description": **null**,

"data": [],

"dataOptions": {

"raw": {

"language": "json"

}

},

"dataMode": "raw",

"headerData": [],

"method": "POST",

"pathVariableData": [],

"queryParams": [],

"auth": **null**,

"events": **null**,

"folder": **null**,

"responses\_order": [],

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"rawModeData": "{\r\n\t\"api\_login\" : \"{{api\_username}}\",\r\n \"api\_password\": \"{{api\_password}}\",\r\n\t\"method\": \"getGameDemo\",\r\n\t\"lang\" : \"{{lang}}\",\r\n\t\"gameid\" : 10497,\r\n\t\"homeurl\" : \"http://url.to.your.page.com\",\r\n\t\"cashierurl\" : \"http://url.to.cashier.page.com\",\r\n\t\"currency\": \"{{currency}}\"\r\n}",

"headers": "",

"pathVariables": {}

},

{

"id": "d39beff6-74f2-4219-836c-148f0043d00c",

"uid": "0-d39beff6-74f2-4219-836c-148f0043d00c",

"name": "credit",

"url": "{{urlstage}}?callerId={{api\_username}}&callerPassword={{api\_password}}&callerPrefix=700ha&action=credit&remote\_id=\"remote ID (player ID), NOT local (player ID)\"&username={{user\_username}}&session\_id=61152976d519b &amount=3&provider=ha&game\_id=10499&game\_id\_hash=ha\_ha-bird-of-thunder&transaction\_id=ha-10bc85c311fceb11b5630050f238e02b&round\_id=10bc85c3-11fc-eb11-b563-0050f238e02b&gameplay\_final=0&is\_freeround\_bet=0&jackpot\_contribution\_in\_amount=0&gamesession\_id=ha\_976201-10499-7be01bbb9be8786f4cc1cd55bd8c8a7e",

"description": **null**,

"data": **null**,

"dataOptions": **null**,

"dataMode": **null**,

"headerData": [],

"method": "GET",

"pathVariableData": [],

"queryParams": [

{

"key": "callerId",

"value": "{{api\_username}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "callerPassword",

"value": "{{api\_password}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "callerPrefix",

"value": "700ha",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "action",

"value": "credit",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "remote\_id",

"value": "\"remote ID (player ID), NOT local (player ID)\"",

"equals": true,

"description": "",

"enabled": true

},

{

"key": "username",

"value": "{{user\_username}}",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "session\_id",

"value": "61152976d519b ",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "amount",

"value": "3",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "provider",

"value": "ha",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "game\_id",

"value": "10499",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "game\_id\_hash",

"value": "ha\_ha-bird-of-thunder",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "transaction\_id",

"value": "ha-10bc85c311fceb11b5630050f238e02b",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "round\_id",

"value": "10bc85c3-11fc-eb11-b563-0050f238e02b",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "gameplay\_final",

"value": "0",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "is\_freeround\_bet",

"value": "0",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "jackpot\_contribution\_in\_amount",

"value": "0",

"equals": true,

"description": **null**,

"enabled": true

},

{

"key": "gamesession\_id",

"value": "ha\_976201-10499-7be01bbb9be8786f4cc1cd55bd8c8a7e",

"equals": true,

"description": **null**,

"enabled": true

}

],

"auth": **null**,

"events": [

{

"listen": "prerequest",

"script": {

"id": "75b2b884-5e42-4826-9997-bf8a8ab6409e",

"exec": [

"var str\_2 = pm.environment.get['api\_password'];\r",

"// Use the CryptoJS\r",

"var mdpassword = CryptoJS.MD5(str\_2).toString();\r",

"// Set the new environment variable\r",

"var a = postman.setEnvironmentVariable('api\_password', mdpassword);\r",

"\r",

"console.log(a);"

],

"type": "text/javascript"

}

}

],

"folder": **null**,

"responses\_order": [],

"preRequestScript": "var str\_2 = pm.environment.get['api\_password'];\r\n// Use the CryptoJS\r\nvar mdpassword = CryptoJS.MD5(str\_2).toString();\r\n// Set the new environment variable\r\nvar a = postman.setEnvironmentVariable('api\_password', mdpassword);\r\n\r\nconsole.log(a);",

"tests": **null**,

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"headers": "",

"pathVariables": {}

},

{

"id": "f35ac06f-7081-477f-97b8-b28d83ace481",

"uid": "0-f35ac06f-7081-477f-97b8-b28d83ace481",

"name": "getGame",

"url": "{{urlstage}}",

"description": **null**,

"data": [],

"dataOptions": {

"raw": {

"language": "json"

}

},

"dataMode": "raw",

"headerData": [],

"method": "POST",

"pathVariableData": [],

"queryParams": [],

"auth": **null**,

"events": **null**,

"folder": **null**,

"responses\_order": [],

"currentHelper": **null**,

"helperAttributes": **null**,

"collectionId": "eac8aebc-3967-4e6d-b2c6-16258f002cd4",

"rawModeData": "{\r\n\"api\_password\": {{api\_password}},\r\n\"api\_login\": {{api\_username}},\r\n\"method\": \"getGame\",\r\n\"lang\": {{lang}},\r\n\"user\_username\": {{user\_username}},\r\n\"user\_password\": {{user\_password}},\r\n\"gameid\": \"ha\_ha-bird-of-thunder\",\r\n\"play\_for\_fun\": 0,\r\n\"currency\": {{currency}}\r\n}",

"headers": "",

"pathVariables": {}

}

]

}

],

"environments": [

{

"id": "44e05778-2f4f-4e7d-8cd0-785fdb2fd517",

"name": "Stage Environments",

"values": [

{

"key": "urlstage",

"value": "https://stage.game-program.com/api/seamless/provider",

"type": "default",

"enabled": true

},

{

"key": "api\_username",

"value": "",

"type": "default",

"enabled": true

},

{

"key": "api\_password",

"value": "",

"type": "default",

"enabled": true

},

{

"key": "currency",

"value": "EUR",

"type": "default",

"enabled": true

},

{

"key": "lang",

"value": "en",

"type": "default",

"enabled": true

},

{

"key": "endpoint",

"value": "",

"type": "default",

"enabled": true

},

{

"key": "user\_username",

"value": "",

"type": "default",

"enabled": true

},

{

"key": "user\_password",

"value": "",

"type": "default",

"enabled": true

}

]

},

{

"id": "f03ce856-b04b-4ea1-b1f9-e46a333a3433",

"name": "My Workspace - globals",

"values": []

}

],

"headerPresets": [],

"globals": []

}

First you choose the document according to what kind you are using. Insert this document into your software tool.

Instructions for Insomnia

|  |
| --- |
|  |
|  |

Instructions for Postman

How to set up the variables in a project

Insomnia

**Game Launch**

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Nov 10, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172171&selectedPageVersions=42&selectedPageVersions=43) by [Nejc Kodele](https://blueoceangaming.atlassian.net/wiki/people/62e78487a41ecbd0ba16ffae?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172171/www.atlassian.com)

Retrieves the games enabled for your agent.

Please note that this list can change over time and is different between staging and production servers! It's best practice to load this list dynamically so you always get the latest games. Please use id\_hash for your game library, because it is the same between staging and production. Id will be different between staging and production.

***retrieved list example*** Expand source

**REST REQUEST**

|  |
| --- |
| //get game list - When you copy the code from here, you need to delete the comments (char: //) !!!  {      "api\_password" : "xapitest",      "api\_login" : xapitest      "method" : "getGameList"      "show\_systems" : 0,      "show\_additional" : false, //only true or false.... (0 or 1 does not work)      "currency" : "EUR"  } |

**REST RESPONSE**

|  |
| --- |
| {"error":0,"response":[{id:2817,name:Example\u2122,type:video-slots,subcategory:example,details:,  new:0,system:mi,position:4,category:SLOTS1,  image:https:\/\/www.example.com\/images\/slots\/small\/mi\/mi-example.png,  image\_preview:https:\/\/www.example.com\/media\/images\/popup\/mi-example.jpg,  mobile:true,play\_for\_fun\_supported:true},...      # REST RESPONSE on show\_systems = 1:  {"error":0,"response":[{"id":"775","name":"Pirates","category":"bgaming",type:"video-slots","subcategory":"other","details":"{\"minbet\":\"0.01\",\"maxbet\":\"75.00\",  \"reels\":\"5\",\"lines\":\"15\",\"freespins\":\"yes\",\"bonusgame\":\"no\"}","new":"0","position":"5","image":"http:\/\/dev.example.com:8090\/media\/images\/slots\  /small\/la\/pirates.png","image\_preview":"http:\/\/dev.example.com:8090\/media\/images\/popup\/pirates.jpg","provider":"la","provider\_name":"Casino Name"}    #REST RESPONSE on "show\_additional" : true:  #Another field named "additional" is added with additional games data:  "error": 0,"response": [ {"id": "10497","name": "Jungle Rumble", ... "gamename": "Jungle Rumble", "report": "ha", "additional": {"aspect\_ratio": "16:9","width": "1280","height": "720","scale\_up": true,"scale\_down": true,"stretching": false,"html5": true,"volatility": "high","max\_exposure": "16200"},"id\_hash": "ha\_ha-jungle-rumble", ... ],  "response\_provider\_logos": {"livecasino": [{"name": "AsiaGaming","system": "ag","image\_black": "https:\/\/stage.game-program.com\/media\/images\/providersLogos\/black\_logos\/ag-dark.png","image\_white": "https:\/\/stage.game-program.com\/media\/images\/providersLogos\/white\_logos\/ag-light.png","image\_colored": "https:\/\/stage.game-program.com\/media\/images\/providersLogos\/org\_logos\/ag.png"},{"name": "BetGames","system": "be","image\_black": "https:\/\/stage.game-program.com\/media\/images\/providersLogos\/black\_logos\/be-dark.png","image\_white": "https:\/\/stage.game-program.com\/media\/images\/providersLogos\/white\_logos\/be-light.png","image\_colored": "https:\/\/stage.game-program.com\/media\/images\/providersLogos\/org\_logos\/be.png"},{"name": "Evolution Livecasino","system": "es","image\_black": "https:\/\/stage.game-program.com\/media\/images\/providersLogos\/black\_logos\/es-dark.png","image\_white": "https:\/\/stage.game-program.com\/media\/images\/providersLogos\/white\_logos\/es-light.png","image\_colored": "https:\/\/stage.game-program.com\/media\/images\/providersLogos\/org\_logos\/es.png"}],... |

Info provided for each game:

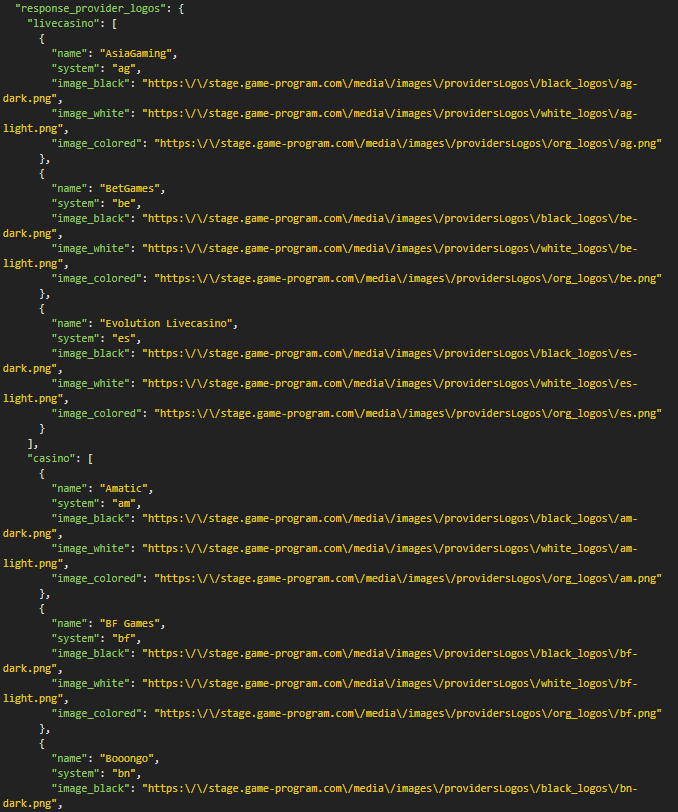
| **Type** | **Parameter** | **Description** | **Usage** |
| --- | --- | --- | --- |
| int | id | gameId to be used with getGame function |  |
| string | name | game name |  |
| string | type | game type:  (livecasino, poker, virtual-sports, sportsbook, live-casino-table, video-slots, table-games, video-poker, virtual-games, scratch-cards, video-bingo, tournaments, livegames, crash-games, fast-games) |  |
| string | subcategory | optional subcategory, name of provider |  |
| string | details | optional data for a game in a json format |  |
| boolean | new | set when the game is new in our system |  |
| string | system | internal shortname for game provider |  |
| string | position |  |  |
| string | category | game main category |  |
| boolean | mobile | specifies if game is mobile |  |
| string | id\_hash | unique string identifier |  |
| int | id\_parent | parent game id of the current game if available |  |
| string | id\_hash\_parent | same as id\_parent but identified as unique string identifier |  |
| boolean | freerounds\_supported | specifies if game supports activation of free rounds via 'addFreeRounds' method |  |
| boolean | featurebuy\_supported | specifies if game supports activation of game features via 'addFeatureBuy' method |  |
| boolean | has\_jackpot | specifies if game has jackpots |  |
| boolean | play\_for\_fun\_supported | set when the game supports play for fun mode |  |
| string | image | game thumbnail image |  |
| string | image\_preview | game screenshot image |  |
| string | image\_filled | direct url to the non-transparent rectangle image icon |  |
| string | image\_portrait | direct url to the portrait image (500 x 800) |  |
| string | image\_square | direct url to the square image (500 x 500) |  |
| string | image\_background | direct url to the game background image |  |
| string | image\_bw | direct url to the black and white image |  |
| string | currency | players currency code |  |

Additional game details (when applicable)

| **Type** | **Name** | **Description** |
| --- | --- | --- |
| string | aspect\_ratio | Aspect ratio of the game as provided from game provider |
| string | width | default width |
| string | height | default height |
| boolean | scale\_up | whether the game scales up |
| boolean | scale\_down | whether the game scales down |
| boolean | stretching | whether the game stretches to fit the screen/iframe |
| boolean | html5 | if the game technology is html5 |
| string | volatility | game volatility (low, medium, high) |
| string | max\_exposure | maximum bet multiplier players can win in the game |
| boolean | megaways | If game is megaways game |
| boolean | bonusbuy | If game supports buying bonus features. |
| string | jackpot\_type | type of jackpot (Non-Jackpot, Non-Progressive,...) |

response\_provider\_logos

if we set parameter show\_additional = true, it is displayed under array: response, displays an array (response\_provider\_logos) where the logos of all providers are located.



show\_systems

if we set parameter show\_systems = 1, additional parameters are displayed in response:

provider -> abbreviation of provider (example: "ha")  
provider\_name (example: "habanero")



**Source Code**

|  |
| --- |
| public function getGameList($show\_systems = false){      return $this->client->getGameList($show\_systems);  } |

Please note that this list can change over time and is different between staging and production servers! It's best practice to load this list dynamically so you always get the latest games. We recommend to call it once a day. Slot popularity is determined by "position" (lower is better).

**1.2 getGame() \***

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Aug 26, 2024](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172237&selectedPageVersions=20&selectedPageVersions=21) by [David Klemenc](https://blueoceangaming.atlassian.net/wiki/people/5ed4bb05d385dd0c267ee1f3?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172237/www.atlassian.com)

Retrieves the desired game embed url for the gameId provided.

*public getGame( int $gameid [, string $lang = string(2) "en" , string $play\_for\_fun = bool FALSE ] )*

**Parameters**

* int**$gameid**required - GameId you recieve from getGameList()
* string**$lang**= string(2) "en" - Lang 2 char language string, for example 'en' or 'cn', not all languages are supported
* string**$play\_for\_fun**= bool FALSE - Play\_for\_fun open game in play for fun mode
* string **$homeurl** - home/back button

**Return Values**

* string - Containing the embed url

| **Name** | **Value** | **Description** |
| --- | --- | --- |
| gamesession\_id | string (255 characters) | enable tracking the player game sessions that is unique per game launch |
| sessionid | string (255 characters) | enable tracking of player sessions that is unique per login |

**REST REQUEST**

|  |
| --- |
| Array  (      [api\_password] => xapitest      [api\_login] =>  xapitest      [method] => getGame      [lang] => en      //[user\_id] => 123 //DEPRECATED      [user\_username] => domendomen2 //not required for fun mode      [user\_password] => 687ee8d988b2411bfc6983c4fca2f6c622819ac8 //not required for fun mode      [gameid] => 568 // you can also use game hash from getGameList to start a game - for example ne#ne-jingle-spin      [homeurl] => http://url.to.your.page.com      [cashierurl] => http://url.to.cashier.page.com //optional      [play\_for\_fun] => 0 //to launch sportsbook in demo use method getGameDemo      [currency] => EUR  ) |

**REST RESPONSE**

|  |
| --- |
| {      "error": 0,      "response": "<https://www.game-program.com/casino/frontend/EmbeddedGame/EmbeddedGame.swf?gameId=99&amp;gameSWF=thePiramid.swf&amp;sessionHash=test> ",      "new\_parameter\_example": "value\_123",      "gamesession\_id": "gs9834ndasd",      "sessionid": "123-898sdjkaERRA"  } |

**SOAP RESPONSE**

|  |
| --- |
| // returned url is to be embedded in this manner, or in a popup window   `iframe src="[http://www.example.com/"](http://www.example.com/%22)` |

**Source Code**

|  |  |
| --- | --- |
| public function getGame($gameid,$lang = 'en',$play\_for\_fun = false){      if(is\_numeric($gameid) && $gameid > 0){          return $this->client->getGame($gameid,$lang,$play\_for\_fun);      }  } | |
| **Game launch diagram:** |
|  |

**1.3 getGameDirect() \***

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172218&selectedPageVersions=11&selectedPageVersions=12) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172218/www.atlassian.com)

Retrieves the desired game embed url or embed code for the gameId provided. Only one or another is returned (url or embed\_code), so your code should expect that. On DEV enviroment, you can test embed\_code via the 914 gameid, In that case you will recieve a short youtube html embed code...

You can also use a fixed gameid by combining system and gametitle, for example ne#ne-attraction this way you're not tied to the ID of the game.

*public getGameDirect( int $gameid [, string $lang = string(2) "en" , string $play\_for\_fun = bool FALSE ] )*

**Parameters**

* int **$gameid** required - GameId you recieve from getGameList()
* string **$lang** = string(2) "en" - Lang see getGame doc.
* string **$play\_for\_fun** = bool FALSE - Play\_for\_fun see getGame doc.
* string **$homeurl** - home/back button

**Return Values**

* array - (url => url, embed\_code => embed\_code)

| **Name** | **Value** | **Description** |
| --- | --- | --- |
| gamesession\_id | string (255 characters) | enable tracking the player game sessions that is unique per game launch |
| sessionid | string (255 characters) | enable tracking of player sessions that is unique per login |

**REST REQUEST**

|  |
| --- |
| Array  (      [api\_password] => xapitest      [api\_login] =>  xapitest      [method] => getGameDirect      [lang] => en      //[user\_id] => 123 //DEPRECATED      [user\_username] => domendomen2      [user\_password] => 687ee8d988b2411bfc6983c4fca2f6c622819ac8      [gameid] => 568      [homeurl] => http://url.to.your.page.com      [play\_for\_fun] => 0      [currency] => EUR  ) |

**REST RESPONSE**

|  |
| --- |
| {"error":0,"response":"https:\/\/www.game-program.com\/casino\/frontend\/EmbeddedGame\/EmbeddedGame.swf?gameId=99&gameSWF=thePiramid.swf&sessionHash=test ","new\_parameter\_example":"value\_123","gamesession\_id":"gs9834ndasd","sessionid":"123-898sdjkaERRA"}      OR when embed\_code is available:    {"error":0,"response":{"embed\_code":"<html>..."}} |

**SOAP RESPONE**

|  |
| --- |
| // returned url is to be embedded in this manner, or in a popup window   `iframe src="[http://www.example.com/"](http://www.example.com/%22)` |

**Source Code**

|  |
| --- |
| public function getGameDirect($gameid,$lang = 'en',$play\_for\_fun = false){      if(is\_numeric($gameid) && $gameid > 0){          return $this->client->getGameDirect($gameid,$lang,$play\_for\_fun);      }  } |

**1.4 getGameDemo() \***

Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172303&selectedPageVersions=4&selectedPageVersions=5) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172303/www.atlassian.com)

Only supported by sportsbooks. Retrieves the desired game embed url for the gameId provided. Launches the sportsbook in demo mode.

**REST REQUEST**

|  |
| --- |
| Array  (      [api\_password] => xapitest      [api\_login] =>  xapitest      [method] => getGameDemo      [lang] => en      [gameid] => 568 // you can also use game hash from getGameList to start a game - for example ds#ds-sportsbook      [homeurl] => http://url.to.your.page.com      [cashierurl] => http://url.to.cashier.page.com //optional      [currency] => EUR  ) |

**1.5 getHiddenList**

Owned by [Klemen Odar](https://blueoceangaming.atlassian.net/wiki/people/5ef084ca66f5de0ab709d6fe?ref=confluence&src=profilecard)

Jul 16, 2024

1 min read

This is the same type of request as **getGameList()** but instead of enabled games it returns the hidden/disabled games and also reason for hiding/disabling it.  
  
**Request example:**

{

    "api\_password" : "xapitest", ← use your api password

    "api\_login" : "xapitest", ← use your api username

    "method" : "getHiddenList",

    "show\_systems" : 0,

    "currency" : "EUR"

}

**Response example:**  
{  
"id": "153495",  
"name": "Lunar New Year Baccarat",  
"type": "live-casino-table",  
"subcategory": "\_bombayLive",  
"parent": "0",  
"details": "{}",  
"new": "1",  
"ext": "png",  
"title": "ol-lunar-new-year-baccarat",  
"system": "ol",  
"position": "2",  
"category": "Bombay Live",  
"frb\_enabled": "0",  
"jackpot": "0",  
"description": "{\"text\_en\":\"\"}",  
"licence": "",  
"plays": "0",  
"rtp": "0",  
"wagering": null,  
"gamename": "",  
"report": "ol",  
"hide": "1",  
"mobile": false,  
"featurebuy\_enabled": "0",  
"release\_date": "2023-01-19",  
"show\_date": null,  
**"hide\_date": "2024-04-19",**  
"lottie": null,  
**"hiding\_reason": "eol",**  
"id\_hash": "ol\_ol-lunar-new-year-baccarat",  
"id\_parent": "0",  
"id\_hash\_parent": "",  
"freerounds\_supported": false,  
"featurebuy\_supported": false,  
"has\_jackpot": false,  
"provider": "ol",  
"provider\_name": "Bombay Live",  
"play\_for\_fun\_supported": true

}

Please keep in mind this is a relatively new addition and we are constantly updating the hiding reasons for the games we have disabled, so some games might still be hidden with the hiding reason set to null.

**Player Management**

**2.1 createPlayer() \***

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Jun 17, 2024](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172289&selectedPageVersions=23&selectedPageVersions=24) by [Klemen Odar](https://blueoceangaming.atlassian.net/wiki/people/5ef084ca66f5de0ab709d6fe?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172289/www.atlassian.com)

Create player using the data passed in the constructor.

Please do not use real player passwords. Because if the player changes his password, you will not be able to log him in via the API. Send a constant password for all players, or create a password by hashing some constant user details (like username) and send that instead.

Username must be min 4 and max 16 characters long! If your player usernames exceed that limit, then please send your player's ID as username!  
Username must not contain any special signs, only upper case letters, lowercase letters, numeric values, minus sign, underscore and dot are allowed!  
Same goes for the nickname parameter!  
First name must contain min 2 and max 25 characters, same goes for last name parameter!  
Once player is created it is not possible to change its currency!

**REST REQUEST (Array)**

|  |
| --- |
| (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => createPlayer      //[user\_id] => 123 //DEPRECATED      [user\_username] => domendomen2 //should be unique - you can use your internal ID for this parameter      [user\_password] => 687ee8d988b2411bfc6983c4fca2f6c622819ac8      [user\_nickname] => notuniquenickname //optional - non unique nickname of a player that is showed in some providers. If not passed user\_username is used      [currency] => EUR  ) |

**REST RESPONSE**

**ON ERROR**

|  |
| --- |
| {     "error":1,     "message":"Player already exists"  } |

**ON SUCCESS**

|  |
| --- |
| {     "error":0,     "response":{        "id":41457, //remote\_id that will be sent in seamless requests        "username":"xxdomendomen2",        "balance":"20.00",        "currencycode":"EUR",        "created":"2013-02-15 14:57:47",        "agent\_balance":null     }  } |

**Source Code**

|  |
| --- |
| public function createPlayer(){      if($this->playerExists() === FALSE){          return $this->client->createPlayer($this->user);      }else{          return "player already exists";      }  } |

**2.2 playerExists() \***

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Jun 17, 2024](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172190&selectedPageVersions=12&selectedPageVersions=13) by [Klemen Odar](https://blueoceangaming.atlassian.net/wiki/people/5ef084ca66f5de0ab709d6fe?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172190/www.atlassian.com)

Check if player exist using the username from the constructor.

**REST REQUEST**

|  |
| --- |
| Array  (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => playerExists      [user\_username] => domendomen2      [currency] => EUR  ) |

**REST RESPONSE**

|  |
| --- |
| OK:  {     "error":0,     "response":{        "id":41457, //remote\_id that will be sent in seamless request        "username":"xxdomendomen2",        "balance":"20.00",        "currencycode":"EUR",        "created":"2013-02-15 14:57:47",        "agent\_balance":"0.25"     }  }    DOESN'T EXIST:  {"error":0,"response":false} |

**Source Code**

|  |
| --- |
| public function playerExists(){    //returns FALSE on no user, else returns details      return $this->client->checkPlayer($this->user['username']);  } |

**4) loginPlayer()**

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Jun 17, 2024](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172185&selectedPageVersions=9&selectedPageVersions=10) by [Klemen Odar](https://blueoceangaming.atlassian.net/wiki/people/5ef084ca66f5de0ab709d6fe?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172185/www.atlassian.com)

Login player using the data passed in the constructor.

**REST REQUEST**

|  |
| --- |
| Array  (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => loginPlayer      //[user\_id] => 123 //DEPRECATED      [user\_username] => domendomen2      [user\_password] => 687ee8d988b2411bfc6983c4fca2f6c622819ac8      [currency] => EUR  ) |

**REST RESPONSE**

|  |
| --- |
| {     "error":0,     "response":{        "id":41457,        "username":"xxdomendomen2",        "balance":"20.00",        "currencycode":"EUR",        "created":"2013-02-15 14:57:47",        "agent\_balance":null,        "sessionid": "5b1f7f74c7793"     }  } |

**Source Code**

|  |
| --- |
| public function loginPlayer(){            // $player Array('username' => '', 'passwrd' => '')      return $this->client->playerLogin($this->user);  } |

**4.1) logoutPlayer()**

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Jun 17, 2024](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172376&selectedPageVersions=8&selectedPageVersions=9) by [Klemen Odar](https://blueoceangaming.atlassian.net/wiki/people/5ef084ca66f5de0ab709d6fe?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172376/www.atlassian.com)

Logout player using the data passed in the constructor.

**REST REQUEST**

|  |
| --- |
| Array  (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => logoutPlayer      [user\_id] => 123      [user\_username] => domendomen2      [user\_password] => 687ee8d988b2411bfc6983c4fca2f6c622819ac8      [currency] => EUR  ) |

**REST RESPONSE**

|  |
| --- |
| {"error":0,"response":  {"id":41457,"username":"xxdomendomen2","balance":"20.00","currencycode":"EUR","created":"2013-02-15  14:57:47","agent\_balance":null}} |

**Source Code**

|  |
| --- |
| public function logoutPlayer(){            // $player Array('username' => '', 'passwrd' => '')      return $this->client->playerLogout($this->user);  } |

**2.5 getSystemUsername()**

Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)

Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172318&selectedPageVersions=8&selectedPageVersions=9) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)

[Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172318/www.atlassian.com)

Checks if system player for given system and player\_id from $player(it's loaded with the user\_username and user\_password) exists and return system player username.

**REST REQUEST**

|  |
| --- |
| Array  (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => getSystemUsername      [user\_username] => domendomen2      [user\_password] => 687ee8d988b2411bfc6983c4fca2f6c622819ac8      [system] => hi      [currency] => EUR  ) |

**REST RESPONSE**

|  |
| --- |
| On error:  1,[error message]      On success:  0,{"info":{"system\_player\_username":"bumbarrr"}} |

**Source Code**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| public function getSystemUsername($system) {}  **2.6 setSystemUsername()**  Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172314&selectedPageVersions=11&selectedPageVersions=12) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172314/www.atlassian.com)  Check if system player for given $splayer\_username, $splayer\_password, $systeem and player\_id from $player(it's loaded with the user\_username and user\_password) exists. If yes then return it otherwise create new system player with provided data and return it.  **REST REQUEST**   |  | | --- | | Array  (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => setSystemUsername      [user\_username] => domendomen2      [user\_password] => 687ee8d988b2411bfc6983c4fca2f6c622819ac8      [splayer\_username] => brad      [splayer\_password] => bradtheking      [system] => hi      [currency] => EUR  ) |   **REST RESPONSE**   |  | | --- | | On error:  1,[error message]      On success:  <br>0,{"info":{"New: system\_player\_username":"thebadmother"}} |   **Source Code**   |  | | --- | | public function setSystemUsername($splayer\_username, $splayer\_password, $system) {} |   **2.7 setSystemPassword()**  Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172306&selectedPageVersions=9&selectedPageVersions=10) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172306/www.atlassian.com)  Updates the system player password with $splayer\_password for given $system.  **REST REQUEST**   |  | | --- | | Array  (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => setSystemPassword      [user\_username] => domendomen2      [user\_password] => 687ee8d988b2411bfc6983c4fca2f6c622819ac8      [splayer\_password] => bradtheking      [system] => hi      [currency] => EUR  ) |   **REST RESPONSE**   |  | | --- | | On error:  1,[error message]      On success:  0,{"info":"System player password was successfully updated!"} |   **Source Code**   |  | | --- | | public function setSystemPassword($splayer\_password, $system) {} |   **7) seamless player balance callback \***   * +5   Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)  Last updated: [Jun 05, 2024](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172336&selectedPageVersions=40&selectedPageVersions=41) by [Alija Šerifović](https://blueoceangaming.atlassian.net/wiki/people/712020:c9b94d06-d197-4ad4-bc33-00b585d0a843?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172336/www.atlassian.com)  Returns players current balance.  **Important info**  We recommended you to do not do any checks on balance requests or keep them to very minimal (salt check is required). Since balance requests can come without session or game\_id in some cases (provider asks for balance before creating session). We suggest to save the player remote id from createPlayer response on your side and then return balance always for that id.  **Request common parameters**   | **Query string name** | **type (max)** | **description** | **mandatory** | | --- | --- | --- | --- | | callerId | String | Caller authentication username | Y | | callerPassword | String | Caller authentication password | Y | | callerPrefix | String | 2 or more letter prefix that is used internally to prefix player username for your account | N | | username | String | Player username (non-prefixed) created by your createPlayer call | Y |   **Player balance**   | **Query string name** | **type (max)** | **description** | **mandatory** | | --- | --- | --- | --- | | action | string | External service action [**balance**, debit, credit] | Y | | remote\_id | integer(10) | Unique player id | Y | | game\_id | string (32 - 40) | Unique game id, see action getGameList | N | | session\_id | string (32 - 40) | Unique game session id generated by gamehub | Y | | key | string (40) | Request validation cache key -> sha1([SALT KEY]+[QUERY STRING]) | Y | | gamesession\_id | string (255) | Enable tracking the player game sessions | N | | game\_id\_hash | string (100) | Related to introduction of id\_hash from getGameList method, can be used the same way as currently is game\_id used | N | | currency | string(3) | Players currency code. Uppercase letters | Y |   **Request:**   |  | | --- | | <REMOTE URL>/?action=balance&callerId=test&callerPassword=12dar67890123&remote\_id=1&session\_id=12345678901234567890 1324567980abcd&key=38432ff064690c9b03da519d0c685b104545 1c9e&new\_parameter=12345&gamesession\_id=98erf743arka&game\_id\_hash=gs\_gs-texas-rangers-reward |   **Response:**   |  | | --- | | {"status":"200","balance":"300.00"} |   **Status codes**  External service can respond with 2 type of status codes. 200 for success, 500 service error. **Do not respond with anything else but 200** on balances if possible. They are requests of informational nature and rejecting them is a bad practice and leads to bad player experience. Only status of 200 will be accepted as OK by the Xapi Platform server.   | **JSON key** | **type (max)** | **description** | **mandatory** | | --- | --- | --- | --- | | status | integer(3) | Status code | Y | | balance | double(10,2) | Players balance (after transaction) | Y | | msg | string(255) | Error message | N |   **Example:** {"status":"200","balance":"300.00"}, {"status":"500","msg":"internal error"}  **8) seamless debit players balance callback \***  Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)  Last updated: [Jun 18, 2024](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172092&selectedPageVersions=53&selectedPageVersions=54) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172092/www.atlassian.com)  Returns player balance after debit transaction.  **Explanation**  Debit requests are called when user makes a bet, so it's basically a bet request, which includes the amount used to lower player balance.  **Request common parameters**   | **Query string name** | **type (max)** | **description** | **mandatory** | | --- | --- | --- | --- | | callerId | String | Caller authentication username | Y | | callerPassword | String | Caller authentication password | Y | | callerPrefix | String | 2 or more letter prefix that is used internally to prefix player username for your account | N | | username | String | Player username (non-prefixed) created by your createPlayer call | Y |   **Debit players balance**   | **Query string name** | **type (max)** | **description** | **mandatory** | | --- | --- | --- | --- | | action | string | External service action [balance, **debit**, credit] | Y | | remote\_id | integer(10) | Unique player id | Y | | amount | double(10,2) | Amount to debit from players account | Y | | provider | string(2) | Game provider | Y | | game\_id | string (32 - 40) | Unique game id, see action available\_games | Y | | transaction\_id | string(32-70) | Unique transaction id for current transaction (**In Evolution multiple players playing same table can share transaction\_id - combined with remote\_id it is unique)** | Y | | gameplay\_final | integer(1) | It tells you if the round is finished(1..finished, 0..unfinished) | Y | | round\_id | string(32-70) | Game round id for current running game round | Y | | session\_id | string (32 - 40) | Unique game session id generated by gamehub | Y | | key | string(40) | Request validation cache key | Y | | gamesession\_id | string (255) | Enable tracking the player game sessions, if player has multiple sessions open we use session from the last launched game | Y | | game\_id\_hash | string (100) | Related to introduction of id\_hash from getGameList method, can be used the same way as currently is game\_id used | N | | fee | double(10,2) | Fees related to this transction. Only used in tournament hub. | N | | tip\_in\_amount | double(10,2) | Player left a tip for the dealer -**it is already included in *amount*field**. Only used in Ezugi livecasino. | N | | is\_freeround\_bet | boolean | This bet came from freerounds (usualy amount will be 0) | N | | freeround\_spins\_remaining | integer(3) | Number of freeround spins remaining. Currently not available for all providers | N | | freeround\_id | string(100) | ID of the freerounds. It should be the same as "freeround\_id" paramater in AddFreerounds response | N | | odd\_factor | double(10,2) | odd factor - only used in sportsbooks | N | | jackpot\_contribution\_in\_amount | double(10,6) | Amount of bet that is contributed to the jackpot contribution. Already included in bet. | N | | jackpot\_contribution\_ids | array of alphanumeric ids | Ids' of the jackpots this bet is contributing to - **deprecated** | N | | jackpot\_contribution\_per\_id | array of doubles | Contributions per different jackpot types, in same order as above contribution\_ids. - **deprecated** | N | | currency | string(3) | Players currency code. Uppercase letters | Y |   **Request**   |  | | --- | | <REMOTE URL>/?action=debit&callerId=test&callerPassword=12dar67890123&remote\_id=1&amount=0.3&game\_id=3 &transaction\_id=27&round\_id=123&session\_id=123456789012 345678901324567980abcd &key=49f749364b129d9f91d2bef7dd044a93af0fb676&new\_parameter=12345&gamesession\_id=98erf743arka&game\_id\_hash=gs\_gs-texas-rangers-reward |   **Response**   |  | | --- | | {"status":"200","balance":"300.00"} |   If we get the request with the transaction\_id that was already processed on our side, on a retry, the response of the transaction should be the same as on first request.  For example if transaction\_id = 61385912731123 response was {"status":"200","balance":"105793.30"}, on a retry with same transaction\_id it should be exactly the same response.  **Status codes**  External service can respond with 4 type of status codes. 200 for success, 500 service error and 403 for debit transaction refusal. Only status of 200 will be accepted as OK by the Xapi Platform server.   * status 500 will make the bet to be retried / canceled and rollback will be sent * status 403 will make the bet be refused  | **JSON key** | **type (max)** | **description** | **mandatory** | | --- | --- | --- | --- | | status | integer(3) | status code | Y | | Balance | double(10,2) | Players balance (after transaction) | Y | | msg | string(255) | Error message | N | | transaction\_id | string(8 - 64) | Transaction id | N |   **Example**   |  | | --- | | {"status":"200","balance":"300.00"}, {"status":"403","balance":"0.30","msg":"Insufficient funds"}, {"status":"500","msg":"internal error"} |   **Additional info**  **Jackpot bets**   * all debit requests that are contributing to jackpot pool wil have jackpot\_contribution\_in\_amount > 0   **Flowchart:**   |  | | --- | |  |   **3.4 seamless rollback transaction callback \***  Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)  Last updated: [Jun 05, 2024](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172157&selectedPageVersions=24&selectedPageVersions=25) by [Alija Šerifović](https://blueoceangaming.atlassian.net/wiki/people/712020:c9b94d06-d197-4ad4-bc33-00b585d0a843?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172157/www.atlassian.com)  **Rollbacks a given transaction. Rely only on transaction\_id.**  **Explanation**  Rollback are sent if any timeouts or other issues happened with debit/credit requests or response. For instance, we send you a debit request, you handle the debit request on your side and for some reasons we don't receive a response and the game crashes. A rollback request will be sent for the debit to player receives the balance back.  **Request common parameters**   | **Query string name** | **type (max)** | **description** | **mandatory** | | --- | --- | --- | --- | | callerId | String | Caller authentication username | Y | | callerPassword | String | Caller authentication password | Y | | callerPrefix | String | 2 or more letter prefix that is used internally to prefix player username for your account | N | | username | String | Player username (non-prefixed) created by your createPlayer call | Y |   **Rollback transaction**   | **Query string name** | **type (max)** | **description** | **mandatory** | | --- | --- | --- | --- | | action | string | External service action [balance, debit, credit, **rollback**] | Y | | remote\_id | integer(10) | Unique player id | Y | | amount | double(10,2) | Amount to credit players account | N | | game\_id | string (32 - 40) | Unique game id, see action available\_games | N | | transaction\_id | string (32 - 70) | Id of transaction that is going to be rollbacked | Y | | round\_id | string (32 - 70) | Game round id for current running game round | N | | session\_id | string (32 - 40) | Unique game session id generated by gamehub | Y | | key | string(40) | Request validation cache key | Y | | gamesession\_id | string (255) | Enable tracking the player game sessions | Y | | game\_id\_hash | string (100) | Related to introduction of id\_hash from getGameList method, can be used the same way as currently is game\_id used | N | | currency | string(3) | Players currency code. Uppercase letters | Y |   **Request:** <REMOTE URL>/?action=rollback&callerId=test&callerPassword=12dar67890123&remote\_id=1&amount=0.3&game\_id =3&transaction\_id=27&round\_id=123&session\_id=123456789012 345678901324567980abcd &key=49f749364b129d9f91d2bef7dd044a93af0fb676&new\_parameter=12345&gamesession\_id=98erf743arka&game\_id\_hash=gs\_gs-texas-rangers-reward  **Response:** {"status":"200","balance":"300.00"}  If we get the request with the transaction\_id that was already processed on our side, on a retry, the response of the transaction should be the same as on first request.  For example if transaction\_id = 61385912731123 response was {"status":"200","balance":"300.00","transactionId":"362984"}, on a retry with same transaction\_id it should be exactly the same response.  If customer does not have a transaction on their end, status should be 404.  **Status codes**  External service can respond with 4 types of status codes. 200 for success, 500 service error, 404 transaction not found and 403 for debit transaction refusal. Only status of 200 will be accepted as OK by the Xapi Platform server.   | **JSON key** | **type (max)** | **description** | **mandatory** | | --- | --- | --- | --- | | status | integer(3) | Status code | Y | | Balance | double(10,2) | Players balance (after transaction) | Y | | msg | string(255) | Error message | N |   **Example:** {"status":"200","balance":"300.00",’transaction\_id’:’123456’}, {"status":"403","balance":"0.30","msg":"Insufficient funds"}, {"status":"500","msg":"internal error"}, {"status":"404","msg":"TRANSACTION\_NOT\_FOUND"}  Deep dive into rollback request: - transaction\_id from the request is the transaction that is being rollbacked - don't use the 'amount' from the request, use the amount in the transaction that is being rollbacked - In case that you get the rollback for transaction\_id that does not exists please return status 404 and some message - Rollback should be allowed even if the round\_id is null or game\_id is null or amount is null or "".   | **Rollback flow:** | | --- | |  |   **Game History**  Owned by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  Mar 01, 2023    **4.1 getGameHistory() \***  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172255&selectedPageVersions=12&selectedPageVersions=13) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172255/www.atlassian.com)  Retrieves player sessions per given date / game.  **Parameters**   * string **$api\_login** required * string **$api\_password** required * string **$method** required value '**getGameHistory**' * string**$user\_username** required  - see loginPlayer method * string**$user\_password** required  - see loginPlayer method * string **$game\_id** optional  - In numeric format *1234*, or gamehash format *gs#gs-power-tiger*. * string **$gamesession\_id** optional - string format returned from getGameDirect or getGame call. * string **$provider**optional - string format 2 characters. If gamesession\_id is not provided, then you can pass 'provider' to filter the query. * string**$date\_start**required - In Y-m-d H:I:S format, UTC timezone (Aug. 9 2012 is represented as '2012-08-09 00:00:00') * string**$date\_end** optional- In Y-m-d H:I:S format, UTC timezone (Aug. 9 2012 is represented as '2012-08-09 00:00:00') * string**$return\_format** optional 'data' or 'url'. Default value is 'data' * int **$page\_number** optional when 'data' format is passed, 1 by default. Determines current page returned. * int**$items\_per\_page** optional when 'data' format is passed, 10 by default. Determines how many items returned per page. Maximum value 100. * boolean **$return\_round\_details**optional Defaults to false, if this parameter is sent, gamesession\_id is mandatory. Sends you gameplay information (bet,win) for that session.   Results are limited to 10 sessions. For more sessions use pagination.  Parameters **game\_id**,**gamesession\_id** and **provider**are filters, and will filter out data in the date range between **date\_start** and **date\_end**.  **Return Values**  **REST REQUEST**   |  | | --- | | POST data    Array  (      [api\_login] => xapitests2s      [api\_password] => xapitest      [method] => getGameHistory      [game\_id] => 2159      [gamesession\_id] => pq\_210640-59522613478e2      [user\_username] => test2      [user\_password] => d22a5985fe6183d51a21d6c9ffa6ab3fb4f7321c      [render] => json      [date\_start] => 2016-06-25 07:27:30      [return\_format] => data      [currency] => EUR  ) |   **REST RESPONSE**   |  | | --- | | // returned data example for return\_format url  {"error":0,"response":"http:\/\/test.dev\/players\/gamehistory\/listsessions\/?system=pq&sessionid=210640-59522613478e2&gameId=2159&date\_start=2016-06-25+07%3A27%3A30&return\_format=url&playerId=223136&hash=ec945cb57ceca77dd96de76e6afc0440ef5bc219","return\_format":"url"}    //returned data example for the default return\_format 'data'  {"error":0,"response":[{"time":"2017-06-27 11:32:03","description":"Pq Sportsbook","gameid":"2159","gamehash":"px\_pq-sportsbook","sid":"210640-59522613478e2","url":"http:\/\/test.dev\/players\/gamehistory\/MjQzNTUx"},{"time":"2017-06-27 11:33:03","description":"Pq Sportsbook","gameid":"2159","gamehash":"px\_pq-sportsbook","sid":"210640-59522613478d2","url":"http:\/\/test.dev\/players\/gamehistory\/MjQzNTUz"}],  "return\_format":"data",  "page\_number": 1,  "pages\_total": 10,  "items\_per\_page": 50,  "items\_current\_page": 50,  "items\_total": 500  } |   If you want to receive additional data  **4.2 getRoundHistory() \***       Owned by [Iztok Stopar Ložar (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/5a97bd38cd281029354b0bd2?ref=confluence&src=profilecard)  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172415&selectedPageVersions=6&selectedPageVersions=7) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172415/www.atlassian.com)  Retrieves players round per given round\_id and game\_id  **Parameters**   * string **$api\_login** required * string **$api\_password** required * string **$method** required value '**getRoundHistory**' * string**$user\_username** required  - see loginPlayer method * string**$user\_password** required  - see loginPlayer method * string **$game\_id** required  - In numeric format *1234*, or gamehash format *gs#gs-power-tiger*. * string **$round\_id**required  - round\_id provided to you in credit/debit requests. * string **$currency**optional - If multicurrency is enabled, this is mandatory.   **Return Values**  **REST REQUEST**   |  | | --- | | POST data    Array  (      [api\_login] => xapitests2s      [api\_password] => xapitest      [method] => getRoundHistory      [game\_id] => 3417      [round\_id] => -1105257958      [user\_username] => test2      [user\_password] => d22a5985fe6183d51a21d6c9ffa6ab3fb4f7321c      [currency] => EUR  ) |   **REST RESPONSE**   |  | | --- | | // returned data example for return\_format url  {    "error": 0,    "response": "http:\/\/dev.pantaloo.com\/players\/tpl\/twoiframes?first=http%3A%2F%2Fdev.pantaloo.com%2Fplayers%2Fgamehistory%2Fspins%2F%3Fplayerid%3DOTgyMTY3%26s%3DZmEwM2EwLTk4NjE4NDY1%26system%3DZ3M%253D%26gameid%3DMzQxNw%253D%253D%26ticket%3DLTExMDUyNTc5NTg%253D&second=http%3A%2F%2Fdev.pantaloo.com%2Fplayers%2Fgamehistory%2Frounddetail%2Fgs%2F%3Fplayerid%3DOTgyMTY3%26t%3DLTExMDUyNTc5NTg%3D%26gameid%3D105",    "currency": "EUR"  } |   Api will return view with two iframes. First iframe will load round history as it is saved on our system while second iframe will show the history as it is fetched from the game provider. This might vary from provider to provider, since not all providers support fetching it via API. In this case, you would only receive our history. Also make sure all parameters being sent are valid, since if not, processed round history might not be found.  **Reports**  Owned by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  Mar 01, 2023    **5.1 getDailyBalances()**  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172408&selectedPageVersions=6&selectedPageVersions=7) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172408/www.atlassian.com)  Retrieves the player final balances per given date.  Date passed must not be for current day, also, if you wish to have this report available to you, you must contact our support first.  *public getDailyBalances( string $date )*  **Parameters**   * string **$date** required - In Y-m-d format (Aug. 9 2012 is represented as '2012-08-09')   **Return Values**   * array   ***returned data example*** Expand source  **REST REQUEST**   |  | | --- | | Array  (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => getDailyBalances      [date] => 2012-09-19      [currency] => EUR  ) |   **REST RESPONSE**   |  | | --- | | {"error":0,"response":[{"player\_id":"42057","username":"Dejan007","balance":"90.84","date":"2013-09-30","username\_real":"jan007"},{"player\_id":"50781","username":"borutle","balance":"19774.25","date":"2013-09-30","username\_real":"rutle"},{"player\_id":"50783","username":"hristians","balance":"1337.15","date":"2013-09-30","username\_real":"istians"},{"player\_id":"50847","username":"matick123","balance":"96.35","date":"2013-09-30","username\_real":"tick123"},{"player\_id":"50848","username":"borutl1","balance":"1764.81","date":"2013-09-30","username\_real":"rutl1"},{"player\_id":"62259","username":"alberto1","balance":"0.00","date":"2013-09-30","username\_real":"berto1"},{"player\_id":"63483","username":"nonsolobet","balance":"50.00","date":"2013-09-30","username\_real":"nsolobet"},{"player\_id":"64780","username":"unikylo","balance":"316.53","date":"2013-09-30","username\_real":"ikylo"},{"player\_id":"64945","username":"gambee","balance":"497.00","date":"2013-09-30","username\_real":"mbee"},{"player\_id":"76559","username":"demo456","balance":"208.80","date":"2013-09-30","username\_real":"mo456"}]} |   **Source Code**   |  | | --- | | public function getDailyBalances($date){      return $this->client->getDailyBalances($date);  } |   **5.2 getDailyReport()**  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172393&selectedPageVersions=9&selectedPageVersions=10) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172393/www.atlassian.com)  Retrieves the player statistics per given date.  Date passed must not be for current day.  *public getDailyReport( string $date [, int $associateid = integer 0 ] )*  **Parameters**   * string **$date** required - In Y-m-d format (Aug. 9 2012 is represented as '2012-08-09') * int **$associateid** = integer 0 - Associate id for which to retrieve reports, default = 0   **Return Values**   * array   ***returned data example*** Expand source  **REST REQUEST**   |  | | --- | | Array  (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => getDailyReport      [date] => 2012-09-19      [associateid] => 0      [currency] => EUR  ) |   **REST RESPONSE**   |  |  |  | | --- | --- | --- | | {    "error": 0,    "response": {      "casino": {        "982167": {          "id": "4983",          "playerid": "982167",          "date": "2019-07-11",          "bet": 10.8,          "win": 4.71,          "tbet": "0.00",          "twin": "0.00",          "net": 6.09,          "gamesPlayed": 13,          "rake": 0,          "fin": 0,          "jackpotContribution": "0.0000",          "jackpotContributionWin": "0.0000",          "freeWin": "0.00",          "bonusBet": "0.00",          "bonusWin": "0.00",          "tip": "0.00",          "fee": "5.00",          "tournament": 0,          "rbet": "0.00",          "system": "es,bs",          "username": "z09981855",          "username\_real": "981855",          "associateid": "0"        }      },      "poker": [],      "allowed\_systems": [        "es",        "bs"      ],      "each\_system": {        "es": {          "982167": {            "id": "4983",            "playerid": "982167",            "date": "2019-07-11",            "bet": "2.00",            "win": "4.00",            "tbet": "0.00",            "twin": "0.00",            "net": "-2.00",            "gamesPlayed": "2",            "rake": "0.00",            "fin": "0.00",            "jackpotContribution": "0.0000",            "jackpotContributionWin": "0.0000",            "freeWin": "0.00",            "bonusBet": "0.00",            "bonusWin": "0.00",            "tip": "0.00",            "fee": "5.00",            "tournament": "0.00",            "rbet": "0.00"          },          "bs": {          "982167": {            "id": "3689",            "playerid": "982167",            "date": "2019-07-11",            "bet": "0.40",            "win": "0.15",            "tbet": "0.00",            "twin": "0.00",            "net": "0.25",            "gamesPlayed": "4",            "rake": "0.00",            "fin": "0.00",            "jackpotContribution": "0.0000",            "jackpotContributionWin": "0.0000",            "freeWin": "0.00",            "bonusBet": "0.00",            "bonusWin": "0.00",            "tip": "0.00",            "fee": "0.00",            "tournament": "0.00",            "rbet": "0.00"          }        }      }    },    "currency": "EUR"  } | | | |  | **Meaning** | |  | Player ID |   **Source Code**   |  | | --- | | public function getDailyReport($date,$associateid = 0){      return $this->client->getDailyReport($date,$associateid);  } |   **5.3 getDailyReportMulti()**  Owned by [Dani Kumar](https://blueoceangaming.atlassian.net/wiki/people/5a4e4996ba56312917a39360?ref=confluence&src=profilecard)  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172124&selectedPageVersions=2&selectedPageVersions=3) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172124/www.atlassian.com)  Retrieves the player statistics for all currencies, per given date.  Date passed must not be for current day.  *public getDailyReportMulti( string $date [, int $associateid = integer 0 ] )*  **Parameters**   * string **$date** required - In Y-m-d format (Aug. 9 2012 is represented as '2012-08-09') * int **$associateid** = integer 0 - Associate id for which to retrieve reports, default = 0   **Return Values**   * array   ***returned data example*** Expand source  **REST REQUEST**   |  | | --- | | Array  (      [api\_password] => xapitest      [api\_login] => xapitest      [method] => getDailyReportMulti      [date] => 2012-09-19      [associateid] => 0      [currency] => EUR  ) |   **REST RESPONSE**   |  | | --- | | {"error":0,"response":{"casino":{"41457":{"playerid":"41457","date":"2012-09-  19","bet":"14520.00","win":"13102.00","net":"1418.00"}},"poker":{"41457\_rr":{"playerid":"41457","date":"2012-09  -19","bet":"1.00","win":"4.00","rake":"13.00","tournament":"0.00","fin":"0.00","system":"rr"}}}} |   **Source Code**   |  | | --- | | public function getDailyReport($date,$associateid = 0){      return $this->client->getDailyReport($date,$associateid);  } |   **Free Rounds**  Owned by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  Mar 01, 2023    **6.1 addFreeRounds()**   * Owned by [Nenad Rauš (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:022193d9-ef07-4b14-ac7f-d3831f7e469b?ref=confluence&src=profilecard)   Last updated: [Nov 28, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172372&selectedPageVersions=27&selectedPageVersions=28) by [Nejc Kodele](https://blueoceangaming.atlassian.net/wiki/people/62e78487a41ecbd0ba16ffae?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172372/www.atlassian.com)  Gives $players $available free rounds for the $game. They are valid until $validTo.  *public addFreeRounds( [ string $title = string(0) "" , string $players , string $games , string $available , string $validTo , string $validFrom = string(0) "" ] )*  **Parameters**   * string **$title** = string(0) "" - Default = "" * string **$playerids** required - Value of the variable looks like 220650,124144 or 220650. Note that there number are EXAMPLES (playerids on our side → remote ID) * string **$gameids** required - Value of the variable looks like 787,789 or 787. Note that these numbers are EXAMPLES * string **$available** required - how many free rounds. * string **$validTo** = string(0) "" - Default = + 1 week * string **$validFrom** = string(0) "" - Default = "" * string **$betlevel** = string(0) "" - Default = "" - Possible values are: min, mid, max. It defines the cost of the bet during the freerounds (if empty it will set "min" by default)   **Return Values**   * type - String   Example: The value of paramater $playerids can be: 220650,124144,2359283 or just 220650  The same is with the parameter $games  **REST REQUEST**   |  | | --- | | Array      (          [api\_password] => xapitest          [api\_login] => xapitest          [method] => addFreeRounds          [tittle] => freeround 1          [playerids] => "220650,124144"          [gameids] => "787,789"          [available] => 4          [validTo] => 2016-05-11          [validFrom] => ""          [betlevel] => ""          [currency] => EUR      ) |   **REST RESPONSE**  **ON ERROR**   |  | | --- | | {      "error": 1,      "message": "addFreeRounds: [error text]"  } |   **ON SUCCESS**   |  | | --- | | {      "error": 0,      "response": "{\"created\":1,\"freeround\_id\":\"5b8fc08970d5544676009e24\"}",      "currency": "EUR"  } |   **Source Code**   |  | | --- | | public function addFreeRounds($title = '', $players, $games, $available, $validTo, $validFrom = '', $betLevel = ''){      return $this->client->addFreeRounds($title, $players, $games, $available, $validTo, $validFrom, $betLevel);  } |   Deep dive into freerounds  Once player enters the game in which he has freerounds all subsequent debit and credit requests will have param "freeround\_id" with id that is returned in addFreeRounds() response.  There is no way to know actual value of the freeround - usually min is min possible bet in the game, mid 0.5€ - 2€ and max around 4€.  "validFrom" time will start at 00:00 and "validTo" will end at 23:59 server time (CET).  Freerounds will be added for all players from "playerids" that open any game from "gameids". Each player will get freerounds only in the first game that is opened and is in "gameids".    **6.2 getFreeRounds()**  Owned by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=2894888961&selectedPageVersions=8&selectedPageVersions=9)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/2894888961/www.atlassian.com)  Get free rounds for the $game. They are valid until $validTo. You can search by multiple games  *public getFreeRounds( [ string $currency = string(0) "" ,string $freeround\_id , string $games , string $validTo , string $validFrom = string(0) "" ] )*  **Parameters**   * string **$currency**= string(0) "" - Default = "EUR" * string **$freeround\_id**string(0) - Value of the variable looks like 627a500001232103210. * string **$gameids** required - Value of the variable looks like 787(id) or gs#gs-three-kings(title). * string **$validTo** = string(0) "" - Default = + 1 week * string **$validFrom** = string(0) "" - Default = "" * string **$betlevel** = string(0) "" - Default = "" - Possible values are: min, mid, max. It defines the cost of the bet during the freerounds   **Required parameters**: $validTo , $currency  Type of requests   * **If freeround\_id is not set (Returns promotions)**   Shows all freerounds that have been added for a specific game **Example**: "**gameids**": "787", "**validTo**": "2023-03-30", "**validFrom**": "2020-02-24" → Lists all freeroundes added for the game: "787" and between the date "validFrom" and "validTo"   * **If freeround\_id is set (Returns users for a specific promotion)**   Shows all freeround that have been used for a specific freeround\_id and in a specific game  The method return all freerounds that have been added for a specific game and filtered by the freeround\_id parameter  **Example**: "**freeround\_id**": "627a500001232103210", "**gameids**": "787",   "**validTo**": "2023-03-30", "**validFrom**": "2020-02-24" → Lists all freeroundes added for the game: "787", for specific round\_id and between the date "validFrom" and "validTo"  **Request**  **REST REQUEST**   |  | | --- | | Array      (          [api\_password] => xapitest          [api\_login] => xapitest          [method] => getFreeRounds          [freeround\_id] => ""          [gameids] => "787"          [validFrom] => 2022-04-30          [validTo] => 2022-11-30          [betlevel] => "min"          [currency] => EUR      ) |   **Response**  **REST RESPONSE**  If freeround\_id is not set:  **ON SUCCESS**   |  | | --- | | {      "error": 0,      "response": [          {              "freeround\_id": "5e79c76972028844521a8470",              "title": "freeround 1",              "players": [                  "981007",                  "1005203"              ],              "games": [                  "10463",                  "10463"              ],              "available": 4,              "betLevel": "min",              "validFrom": "2020-03-24 01:00:00",              "validTo": "2020-03-31 01:59:59",              "created": "2020-03-24 09:40:09",              "received": 1          },      ...  } |   If freeround\_id is set:  **ON SUCCESS**   |  | | --- | | {      "error": 0,          "response": [              {                  "freeround\_id": "5e79c76972028844521a8470",                  "players": [                      "981007",                  ],                  "games": [                      "10463"                  ],                  "available": 4,                  "betLevel": "min",                  "validFrom": "2020-03-24 01:00:00",                  "validTo": "2020-03-31 01:59:59",                  "created": "2020-03-24 09:40:09",                  "used": 0,                  "activated" null,                  "activatedDate": null,                  "win":null                  "completed":null              },              {                  "freeround\_id": "5e79c76972028844521a8470",                  "players": [                      "981007",                  ],                  "games": [                      "10463"                  ],                  "available": 4,                  "betLevel": "min",                  "validFrom": "2020-03-24 01:00:00",                  "validTo": "2020-03-31 01:59:59",                  "created": "2020-03-24 09:40:09",                  "used": 0,                  "activated" null,                  "activatedDate": null,                  "win":null                  "completed":null              }              ...          ],          "currency" : "EUR"  } |   **ON ERROR**   |  | | --- | | {      "error": 1,      "message": "unknown command"  } |   **Source Code**   |  | | --- | | public function getFreeRounds($freeroundId, $games, $validTo, $validFrom = '', $betLevel = ''){      return $this->client->addFreeRounds($freeroundId, $games, $validTo, $validFrom, $betLevel);  } |   **6.3 removeFreeRounds()**  Owned by [Jurec (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/557058:871f0f52-df40-42a2-9977-5cf6e9d44fca?ref=confluence&src=profilecard)  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=1209172142&selectedPageVersions=11&selectedPageVersions=12) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/1209172142/www.atlassian.com)  **It is not possible to remove freerounds once they are created on game provider side (when player enters the game).**  **Parameters**   * string **$freeround\_id**required string  - You get it in response of addFreeRounds() * string **$playerids**=string - Value of the variable looks like 220650,124144 or 220650. If param is not set it will take all players registered under given freeround\_id. Note that these nubers are EXAMPLES   **Return Values**   * type - String   **REST REQUEST**   |  | | --- | | Array      (          [api\_password] => xapitest          [api\_login] => xapitest          [method] => removeFreeRounds          [playerids] => "220650,124144"          [freeround\_id] => "5bb1ba5a492f591694004d3e"          [currency] => EUR      ) |   **REST RESPONSE**  **ON ERROR**   |  | | --- | | {      "error": 1,      "message": "Gameprovider.removeFreeRounds recieved invalid input:The freeround\_id is not valid."  } |   **ON SUCCESS**   |  | | --- | | {      "error": 0,      "response": {          "successfull\_removals": [              124144          ],          "unsuccessfull\_removals": [              220650          ]      },      "currency": "EUR"  } |   **Feature Buy**  Owned by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  Mar 01, 2023    **7.1 addFeatureBuy()**  Owned by [Iztok Stopar Ložar (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/5a97bd38cd281029354b0bd2?ref=confluence&src=profilecard)  Last updated: [Mar 01, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=3028942849&selectedPageVersions=1&selectedPageVersions=2) by [Anej Hrobat (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/60f9284633876e0068e81b64?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/3028942849/www.atlassian.com)  Gives $players feature buy promotion for the $game. They are valid until $validTo.  *public addFeatureBuy( [ string $title = string(0) "" , string $players , string $games , string $available , string $validTo , string $validFrom = string(0) "" ] )*  **Parameters**   * string **$title** = string(0) "" - Default = "" * string **$playerids** required - Value of the variable looks like 220650,124144 or 220650. Note that there nuber are EXAMPLES * string **$gameids** required - Value of the variable looks like 787,789 or 787. Note that these numbers are EXAMPLES * string **$validTo** = string(0) "" - Default = + 1 week * string **$validFrom** = string(0) "" - Default = "" * string **$betlevel** = string(0) "" - Default = "" - Possible values are: min, mid, max. It defines the cost of the bet during the feature buy. **Note: feature buy promotions can on average be worth between 50-100€ on min bet, so mid and max are worth even more.**   **Return Values**   * type - String   Example: The value of paramater $playerids can be: 220650,124144,2359283 or just 220650  The same is with the parameter $games  **REST REQUEST**   |  | | --- | | Array      (          [api\_password] => xapitest          [api\_login] => xapitest          [method] => addFeatureBuy          [tittle] => featurebuy 1          [playerids] => "220650,124144"          [gameids] => "787,789"          [validTo] => 2016-05-11          [validFrom] => ""          [betlevel] => ""          [currency] => EUR      ) |   **REST RESPONSE**  **ON ERROR**   |  | | --- | | 1,addFeatureBuy: [text error] |   **ON SUCCESS**   |  | | --- | | {      "error": 0,      "response": "{\"created\":1,\"featurebuy\_id\":\"5b8fc08970d5544676009e24\"}",      "currency": "EUR"  } |   **Source Code**   |  | | --- | | public function addFeatureBuy($title = '', $players, $games, $validTo, $validFrom = '', $betLevel = ''){      return $this->client->addFeatureBuy($title, $players, $games, $validTo, $validFrom, $betLevel);  } |   Deep dive into freerounds  Once player enters the game in which he has featurebuy all subsequent debit and credit requests will have param "featurebuy\_id" with id that is returned in addFeatureBuy() response.  There is no way to know actual value of the freeround - usually min is min possible bet in the game, mid next possible bet and max third possible bet.  "validFrom" time will start at 00:00 and "validTo" will end at 23:59 server time (CET).  Featurebuys will be added for all players from "playerids" that open any game from "gameids". Each player will get featurebuy only in the first game that is opened and is in "gameids".  **7.2 getFeatureBuy()**  Owned by [Iztok Stopar Ložar (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/5a97bd38cd281029354b0bd2?ref=confluence&src=profilecard)  Last updated: [May 31, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=3108175873&selectedPageVersions=2&selectedPageVersions=3)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/3108175873/www.atlassian.com)  Get featurebuy rounds for the $game. They are valid until $validTo. You can search by multiple games  *public getFeatureBuy( [ string $currency = string(0) "" ,string $featurebuy\_id , string $games , string $validTo , string $validFrom = string(0) "" ] )*  **Parameters**   * string **$currency**= string(0) "" - Default = "EUR" * string **$featurebuy\_id**string(0) - Value of the variable looks like 627a500001232103210. * string **$gameids** required - Value of the variable looks like 787(id) or gs#gs-three-kings(title). * string **$validTo** = string(0) "" - Default = + 1 week * string **$validFrom** = string(0) "" - Default = "" * string **$betLevel** = string(0) "" - Default = "" - Possible values are: min, mid, max. It defines the cost of the bet during the freerounds   **Required parameters**: $validTo , $currency  Type of requests   * **If freeround\_id is not set (Returns promotions)**   Shows all freerounds that have been added for a specific game **Example**: "**gameids**": "787", "**validTo**": "2023-03-30", "**validFrom**": "2020-02-24" → Lists all freeroundes added for the game: "787" and between the date "validFrom" and "validTo"   * **If featurebuy\_id is set (Returns users for a specific promotion)**   Shows all freeround that have been used for a specific freeround\_id and in a specific game  The method return all featurebuys that have been added for a specific game and filtered by the freeround\_id parameter  **Example**: "**featurebuy\_id**": "627a500001232103210", "**gameids**": "787",   "**validTo**": "2023-03-30", "**validFrom**": "2020-02-24" → Lists all featurebuys added for the game: "787", for specific **featurebuy\_id** and between the date "validFrom" and "validTo"  **Request**  **REST REQUEST**   |  | | --- | | Array      (          [api\_password] => xapitest          [api\_login] => xapitest          [method] => getFeatureBuy          [featurebuy\_id] => ""          [gameids] => "787"          [validFrom] => 2022-04-30          [validTo] => 2022-11-30          [betlevel] => "min"          [currency] => EUR      ) |   **Response**  **REST RESPONSE**  If freeround\_id is not set:  **ON SUCCESS**   |  | | --- | | {      "error": 0,      "response": [          {              "featurebuy\_id": "64074db4163b0000e600515b",              "players": [                  "1007497"              ],              "games": [                  "qs#qs-artemis-vs-medusa",                  "qs#qs-artemis-vs-medusa-mobile"              ],              "betLevel": "min",              "validFrom": "2016-05-01 02:00:00",              "validTo": "2023-04-30 01:59:59",              "created": "2023-03-07 15:44:04",              "received": 0          },          {              "featurebuy\_id": "64075295163b0000e6005173",              "players": [                  "1007497"              ],              "games": [                  "qs#qs-artemis-vs-medusa",                  "qs#qs-artemis-vs-medusa-mobile"              ],              "betLevel": "min",              "validFrom": "2016-05-01 02:00:00",              "validTo": "2023-04-30 01:59:59",              "created": "2023-03-07 16:04:53",              "received": 0          }      ],      "currency": "EUR"  } |   If freeround\_id is set:  **ON SUCCESS**   |  | | --- | | {      "error": 0,      "response": [          {              "featurebuy\_id": "64074db4163b0000e600515b",              "player": "1007497",              "games": [                  "qs#qs-artemis-vs-medusa",                  "qs#qs-artemis-vs-medusa-mobile"              ],              "betLevel": "min",              "validFrom": "2016-05-01 02:00:00",              "validTo": "2023-04-30 01:59:59",              "created": "2023-03-07 15:44:04",              "used": 0,              "activated": null,              "activatedDate": null,              "win": null,              "completed": false          }      ],      "currency": "EUR"  } |   **ON ERROR**   |  | | --- | | {      "error": 1,      "message": "unknown command"  } |   **Source Code**   |  | | --- | | public function getFeatureBuy($featurebuy\_id, $games, $validTo, $validFrom = '', $betLevel = ''){      return $this->client->getFeatureBuy($featurebuy\_id, $games, $validTo, $validFrom, $betLevel);  } |   **7.3 removeFeatureBuy()**  Owned by [Iztok Stopar Ložar (Unlicensed)](https://blueoceangaming.atlassian.net/wiki/people/5a97bd38cd281029354b0bd2?ref=confluence&src=profilecard)  Last updated: [Sept 13, 2023](https://blueoceangaming.atlassian.net/wiki/pages/diffpagesbyversion.action?pageId=3108175885&selectedPageVersions=4&selectedPageVersions=5) by [Jaka Minkuz](https://blueoceangaming.atlassian.net/wiki/people/712020:1796f7ee-3ae0-4ecb-915c-d7986f191545?ref=confluence&src=profilecard)  [Legacy editor](https://blueoceangaming.atlassian.net/wiki/spaces/iGPPD/pages/3108175885/www.atlassian.com)  **It is not possible to remove featurebuys once they are created on game provider side (when player enters the game).**  **Parameters**   * string **$featurebuy\_id**required string  - You get it in response of addFeatureBuy() * string **$playerids**=string - Value of the variable looks like 220650,124144 or 220650. If parameter is not set it will take all players registered under given featurebuy\_id. Note that these numbers are EXAMPLES   **Return Values**   * type - String   **REST REQUEST**   |  | | --- | | Array      (          [api\_password] => xapitest          [api\_login] => xapitest          [method] => removeFeatureBuy          [playerids] => "220650,124144"          [featurebuy\_id] => "5bb1ba5a492f591694004d3e"          [currency] => EUR      ) |   **REST RESPONSE**  **ON ERROR**   |  | | --- | | {      "error": 1,      "message": "Gameprovider.removeFeatureBuy recieved invalid input:The featurebuy\_id is not valid."  } |   **ON SUCCESS**   |  | | --- | | {      "error": 0,      "response": {          "successfull\_removals": [              124144          ],          "unsuccessfull\_removals": [              220650          ]      },      "currency": "EUR"  } | |