Backend API Specification

( v1.0.0 )

**Table Of Contents**

[1. Unauthorize API 3](#_Toc175185518)

[1) Launch game 3](#_Toc175185519)

[2) Create user (Transfer) 4](#_Toc175185520)

[3) Deposit user money (Transfer) 5](#_Toc175185521)

[4) Withdraw user money (Transfer) 5](#_Toc175185522)

[5) Withdraw user money all (Transfer) 6](#_Toc175185523)

[6) User information retrieve (Transfer) 7](#_Toc175185524)

[7) Pull report by date 8](#_Toc175185525)

[8) Pull report by index 10](#_Toc175185526)

[9) Get wager detail 11](#_Toc175185527)

[10) Get game company list 12](#_Toc175185528)

[11) Get game list 13](#_Toc175185529)

[12) Get agent information 14](#_Toc175185530)

[13) Get wager detail url 15](#_Toc175185531)

[2. Authorize API 17](#_Toc175185532)

[1) GetAffiliaterInfo 17](#_Toc175185533)

[2) Change user balance ( ChangeBalance ) 18](#_Toc175185534)

[3) Delivering game details ( UpdateDetail ) 19](#_Toc175185535)

[3. Appendix 21](#_Toc175185536)

[1) Response code 21](#_Toc175185537)

[2) Game type 21](#_Toc175185538)

[3) Language 21](#_Toc175185539)

[4. Model 23](#_Toc175185540)

[1) User 23](#_Toc175185541)

[2) Wager (transaction history) 23](#_Toc175185542)

[3) Vendor (game company information) 23](#_Toc175185543)

[4) VendorGame (Game Information) 23](#_Toc175185544)

1. Unauthorize API
2. GetVendorGames

You must call this API to get vendor game list.

vendorCode must be a code obtained from the GetVendors API.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | GetVendorGames | Y |
| agentCode | String | agent code | Y |
| gameType | Integer | Game type 1-slot, 9- minigame | Y |

* Request example:

{

“method” : “ GetVendorGames”,

“agentCode ” : “ testAgent ”,

“gameType” : 9

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| vendorGames | List<VendorGame> | Game list |

* Response example:
* success

{

"status" : 0,

" msg" : "SUCCESS",

"vendorGames": [

 {

             "vendorName": "{\"en\":\"Spribe Gaming\"}",

            "vendorCode": "mini-spribe",

            "gameType": 9,

            "gameCode": "Aviator",

            "imageUrl": "{\"en\":\"https://app.roogsino.io/resources/image/games/Av-new@2x.png\"}",

            "gameName": "{\"en\":\"Aviator\",\"ko\":\"Aviator\"}"

        }

]

}

* error

{

“status”: 12,

" msg ” : "INVALID\_VENDOR"

}

1. Launch game

You must call this API to launch a game.

vendorCode must be a code obtained from the GetVendors API.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | GetGameUrl | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| userCode | String | Site user code | Y |
| nickname | String | User nickname | N |
| vendorCode | String | Game company code | Y |
| gameCode | String | This is the game code needed to start a specific game.  To get the game code, refer to the 2.11 API and see if there is a value for it. | N |
| currencyCode | String | Currency | Y |
| language | String | Language used (see Appendix 4.3 ) | N |

* Request example:

{

“ method” : “ GetGameUrl ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ userCode ” : “Player1”,

“ nickName ” : “King”,

“ vendorCode ”: “ Evolution\_Casino ”,

“ language ” : “ ko”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| launchUrl | String | Site user code |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

" launchUrl ": "https://evolution.com/entry?JSESSIONID = TVE12E22154G"

}

* error

{

“status”: 12,

" msg ” : "INVALID\_VENDOR"

}

1. Create user (Transfer)

In case of transfer method, this API must be called to create a new user.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | CreateUser | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| userCode | String | Site user code | Y |

* Request example:

{

“ method” : “ CreateUser ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ”: “ testAgent ”,

“ userCode ” : “Player1”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| userCode | String | Site user code |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

" userCode ": "Player1"

}

* error

{

"status" : 7,

" msg " : "DUPLICATE\_USER"

}

1. Deposit user money (Transfer)

In order to deposit money to a user, this API must be called.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | Deposit | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| userCode | String | site user code | Y |
| currencyCode | String | Currency | Y |
| amount | Decimal | Change amount | Y |

* Request example:

{

“ method” : “Deposit”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ userCode ” : “Player1”,

“ amount” : 10000

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| prevBalance | Decimal | User money before change |
| balance | Decimal | Money held by user after change |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

“ prevBalance ” : 100000,

"balance" : 110000

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. Withdraw user money (Transfer)

To withdraw the user's money, you must call this API.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | Withdraw | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| userCode | String | Site user code | Y |
| currencyCode | String | Currency | Y |
| amount | Decimal | Change amount | Y |

* Request example:

{

“ method” : “Withdraw”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ userCode ” : “Player1”,

“amount ”: 10000

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| prevBalance | Decimal | User money before change |
| balance | Decimal | Money held by user after change |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

" prevBalance ” :100000,

"balance" : 90000

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. Withdraw user money all (Transfer)

To withdraw all of the user's money, you must call this API.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | Category | explanation | essential |
| method | String | WithdrawAll | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| userCode | String | Site user code | Y |

* Request example:

{

“ method” : “ WithdrawAll ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ userCode ” : “Player1”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | Category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| prevBalance | Decimal | User money before change |
| balance | Decimal | Money held by user after change |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

“ prevBalance ” : 90000,

“balance” :0

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. User information retrieve (Transfer)

To retrieve user information from Transfer, you must call this API.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | Category | explanation | essential |
| method | String | GetUserInfo | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| userCode | String | Site user code  \*If empty, check all user information | N |

* Request example:

{

“ method” : “ GetUserInfo ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ userCode ” : “Player1”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| users | List<User> | User information  \*Refer to model 5.1 for user details |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

“ users” : [

{

“ userCode ” : ”Player 1”,

“ balance” : 100000

}

]

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. Pull report by date

You must call this API to retrieve the completed transaction history with the last transaction record date .

startDate and endDate exceeds 5 minutes or the time is invalid, an INVALID\_TIME error is returned.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | Category | explanation | essential |
| method | String | ReportByDate | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| startDate | DateTime | Start date and time | Y |
| endDate | DateTime | End date and time | Y |

* Request example:

{

“ method” : “ ReportByDate ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ startDate ” : “2023-06-25 10:00:00”,

“ endDate ” : “2023-06-25 10:05:00”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| wagers | List<Wager> | Transaction information  \*For detailed information on Wager, refer to Model 5.2 |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

“ wager” : [

{

“ userCode ” : “Player1”,

“ vendorCode ” : ” Evolution\_Casino ”,

“ gameType ” : 2,

“ gameCode ” : ”LightningBac0001”,

“ gameRoundId ” : ”16033298890220”,

" wagerId ” : 25641,

“ betAmount ” : 1000,

“ payoutAmount ” : 2000,

“ beforeBalance ” : 10000,

“ afterBalance ” : 11000,

“ detail” : null,

" createdOn ” : "2023-06-25 10:00:12”,

" modifiedOn ” : "2023-06-25 10:00:30” ,

“ settlementOn ” : “2023-06-25 10:00:30”,

" isFinished " : true,

" status" : 1

},

**…**

]

}

* error

{

“status”: 20,

" msg ” : "INVALID\_TIME"

}

1. Pull report by index

You must call this API to search transaction details using the transaction details ID .

When making your first request, please enter 0 in start WagerId , and for the next request, please enter lastWagerId+1.

Among the transaction details received in response, details with isFinished = false must be searched again using the Get wager details API.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | ReportById \_ | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| startWagerId | Long | starting index | Y |
| count | Integer | Number of details | Y |

* Request example:

{

“ method” : “ ReportById ” ,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ start WagerId ” : 25641,

“ count” : 100

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| lastWagerId | Long | Last history ID |
| wagers | List<Wager> | Transaction information  \*For detailed information on Wager, refer to Model 5.2 |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

" lastWagerId " : 25645 ,

“ wager” : [

{

“ userCode ” : “Player1”,

“ vendorCode ” : ” Evolution\_Casino ”,

“ gameType ” : 2,

“ gameCode ” : ”LightningBac0001”,

“ gameRoundId ” : ”16033298890220”,

" wagerId ” : 25641,

“ betAmount ” : 1000,

“ payoutAmount ” : 2000,

“ beforeBalance ” : 10000,

“ afterBalance ” : 11000,

“ detail” : null,

" createdOn " : "2023-06-25 10:00:12”,

" modifiedOn " : "2023-06-25 10:00:30” ,

" settlementOn " : "2023-06-25 10:00:30”,

" isFinished " : true,

" status" : 1

},

**…**

]

}

* error

{

“status”: 1,

“ msg ” : “INTERNAL\_ERROR “

}

1. Get wager detail

You must call this API to retrieve wager detail using the wager ID.

If there is no information to investigate, it returns INVALID\_WAGER(18). If the details are incomplete, the details with isFinished =false are returned.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | GetWagerInfo | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| wagerId | Long | Transaction ID | Y |

* Request example:

{

“ method” : “ GetWagerInfo ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ wagerId ” : 25641

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| wager | Wager | Transaction information  \*For detailed information on Wager, refer to Model 5.2 |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

“ wager” : {

“ userCode ” : “Player1”,

“ vendorCode ” : ” Evolution\_Casino ”, “ gameType ” : 2,

“ gameCode ” : ”LightningBac0001”,

“ gameRoundId ” : ”16033298890220”,

" wagerId ” : 25641,

“ betAmount ” : 1000,

“ payoutAmount ” : 2000,

“ beforeBalance ” : 10000,

“ afterBalance ” : 11000,

“ detail” : null,

" createdOn ” : "2023-06-25 10:00:12”,

" modifiedOn ” : "2023-06-25 10:00:30” ,

“ settlementOn ” : “2023-06-25 10:00:30”,

" isFinished " : true,

" status" : 1

}

}

* error

{

“status”: 18,

" msg ” : "INVALID\_WAGER"

}

1. Get game company list

To view the list of game companies, you must call this API.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | GetVendors | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |

* Request example:

{

“ method” : “ GetVendors ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| vendors | List<Vendor> | List of game companies  \*For vendor details, refer to Model 5.3 |

* Response example :
* success

{

"status" : 0,

" msg " : "SUCCESS",

“ vendors” : [

{

“ vendorCode ” : “ Evolution\_Casino “,

“ vendorName ” : ”Evolution”,

“ gameType ” : 2

},

**…**

]

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. Get game list

You must call this API to view the game list of a specified game company.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | GetVendorGames | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| vendorCode | String | game company code | Y |

* Request example:

{

“ method” : “ GetVendorGames ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ vendorCode ” : “ Pragmatic\_Slot ”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| vendorGames | List<VendorGame> | List of game companies  \* For detailed information on VendorGame , refer to Model 5.4 |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

“ vendorGames ” : [

{

" gameCode ": "vs20olympgate",

" gameName ": "{\" en \":\"Gate of Olympus\", \" en \":\"Gates of Olympus\"}",

" gameType ": 1,

" imageUrl ": "{\"en\ ":\ "https://api-sg57.ppgames.net/game\_pic/square/200/vs20olympgate.png\"}"

}, **…**

]

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. Get agent information

To retrieve user information from Transfer, you must call this API.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | GetAgentInfo | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |

* Request example:

{

“ method” : “ GetAgentInfo ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| balance | Decimal | Agent's money |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

“ balance” : 1000000

}

* error

{

“status”: 3,

" msg ” : "INVALID\_AGENT"

}

1. Get wager detail url

To retrieve wager details using the wager ID, you must call this API.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | GetDetailUrl | Y |
| token | String | agent token | Y |
| agentCode | String | agent code | Y |
| wagerId | Long | Wager ID | Y |

* Request example:

{

“ method” : “ GetDetailUrl ”,

“ token” : “s3dcvdfeff4f889765xfd5123”,

“ agentCode ” : “ testAgent ”,

“ wagerId ” : 25641

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| detailUrl | String | Transaction details URL |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

" detailUrl " : " https://api.xyz.com / GameDetail?token = s3dcvdfeff4f889765xfd5123 &wagerId= 25641"

}

* error

{

"status": 18,

" msg " : "INVALID\_WAGER"

}

1. Authorize API

* For authorize API integration, you must put token to header below.
* const response = await fetch(`${backendUrl}/backend/authorizeapi`, {
* method: "POST",
* headers: {
* "X-Access-Token": accessToken,
* "Content-Type": "application/json",
* },
* body: JSON.stringify({
* method: "CreateAffiliater",
* affiliaterCode: affiliateCode,
* }),
* });

1. CreateAffiliater

You calls this API to create affiliater.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | CreateAffiliater | Y |
| affiliaterCode | String | Affiliater code | Y |

* Request example:

{

“method” : “ CreateAffiliater”,

“affiliaterCode” : “xyz”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |

* Response example:
* success

{

"status" : 0,

"msg" : "SUCCESS"

}

* error

{

“status”: 1,

" msg ” : "Already exist"

}

1. GetAffiliaterInfo

You calls this API for get information for affiliate page.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | GetAffiliaterInfo | Y |
| currencyCode | String | Currency code | Y |

* Request example:

{

“method” : “ GetAffiliaterInfo”,

“currencyCode” : “ROOG”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| affiliateCodes | List<String> |  |
| lastSettleTime | DateTime |  |
| currencyCode | String |  |
| totalBetCount | Integer |  |
| totalBetAmount | Decimal |  |
| totalPayoutAmount | Decimal |  |
| totalIncome | Decimal |  |
| totalReferralCount | Integer |  |
| referralInfos | List<ReferralInfo> |  |

* Response example:
* success

{

"status" : 0,

"msg" : "SUCCESS",

"affiliateCodes" : [“xyz”],

"currencyCode" : “ROOG”,

“totalBetCount”:100,

“totalBetAmoutn”:12000.56,

“totalPayoutAmount”:13000,

“totalIncome”:20.6,

“totalReferralCount”:3,

“refferalInfos”:[

{

“affiliaterCode”:”xyz”,

“userCode”:”serrat”

“betCount”:2,

“betAmount”:60,

“payoutAmount”:100,

“income”:0.5

}

]

}

* error

{

“status”: 5,

" msg ” : "No affiliater"

}

1. ClaimRefferalBonus

You calls this API to claim referral bonus.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | ClaimRefferalBonus | Y |
| currencyCode | String | Currency | Y |

* Request example:

{

“method” : “ CreateAffiliaterCode”,

“currencyCode” : “ROOG”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| bonus | Decimal |  |

* Response example:
* success

{

"status" : 0,

"msg" : "SUCCESS",

“bonus”:5

}

* error

{

“status”: 1,

" msg ” : "Already exist"

}

1. GetUnreadNoticeList

You calls this API to get unread notice list.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| field | category | explanation | essential |
| method | String | GetUnreadNoticeList | Y |

* Request example:

{

“method” : “ GetUnreadNoticeList”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| data | List<Notice> |  |

* Response example:
* success

{

"status" : 0,

"msg" : "SUCCESS",

“data”:[

{

“Id”:1234,

“Title”:”Deposit bonus”,

“CreatedAt”:” 2024-08-29 10:00:12”

}

]

}

* error

{

“status”: 1,

" msg ” : "Already exist"

}

1. ReadNotice

You calls this API to get notice and read.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Field | category | explanation | essential |
| method | String | ReadNotice | Y |
| Id | Integer |  | Y |

* Request example:

{

“method” : “ReadNotice”,

“id”:1234

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| Field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| data | Notice |  |

* Response example:
* success

{

"status" : 0,

"msg" : "SUCCESS",

“data”:[

{

“Id”:1234,

“AgentCode”:”serrat”,

“UserCode”:”123123123”,

“Title”:”Deposit bonus”,

“Content”:” You received a referral game play bonus between January 1st and January 5th”

“CreatedAt”:” 2024-08-29 10:00:12”

}

]

}

* error

{

“status”: 1,

" msg ” : "Already exist"

}

1. CheckBalance

You calls this API to check balance.

This API is used in Solana chain

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Field | category | explanation | essential |
| method | String | CheckBalance | Y |
| chain | String |  | Y |
| coinType | String |  | Y |

* Request example:

{

“method” : “CheckBalance”,

“chain”:”Solana”,

“coinType”:”ROOG”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| Field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |

* Response example:
* success

{

"status" : 0,

"msg" : "SUCCESS",

}

* error

{

“status”: 1,

" msg ” : "Fail"

}

1. GetBalanceModalInfo

When user launch Balance modal dialog, you calls this API to get modal information.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Field | category | explanation | essential |
| Method | String | GetBalanceModalInfo | Y |
| coinType | String | Coin type | Y |

* Request example:

{

“ method” : “ GetBalanceModalInfo”,

“ coinType” : “xrpl”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| balance | Decimal | User's money |
| depositAddress | String | Deposit address |
| modalMessage | String | Message shown in modal |
| withdrawalMaxLimit | Decimal | Withdrawl max limit |
| depositMinLimit | Decimal | Minimum deposit limit |
| pointConvertRatio | Dictionary | If coinType is virtual |
| pointConvertLimit | Dictionary | If coinType is virtual |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

"balance" : 10000,

“depositAddress”:”orai111111”,

“modalMessage”:”The first deposit amount must be larger than 10 xrpl”,

“withdrawalMaxLimit”:30, //10% of bet amount,

“depositMinLimit”:10xrpl,

“pointConvertRatio”:{

“sol”:0.5,

“USDC”:1

},

“pointConvertLimit”:{ //1% of bet amount

“sol”:1,

“USDC”:100

}

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. GetVirtualBalanceModalInfo

When user launch Balance modal dialog, you calls this API to get modal information.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Field | category | explanation | essential |
| Method | String | GetBalanceModalInfo | Y |
| coinType | String | Coin type | Y |

* Request example:

{

“ method” : “ GetBalanceModalInfo”,

“ coinType” : “xrpl”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| balance | Decimal | User's money |
| depositAddress | String | Deposit address |
| modalMessage | String | Message shown in modal |
| withdrawalMaxLimit | Decimal | Withdrawl max limit |
| depositMinLimit | Decimal | Minimum deposit limit |
| pointConvertRatio | Dictionary | If coinType is virtual |
| pointConvertLimit | Dictionary | If coinType is virtual |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

"balance" : 10000,

“depositAddress”:”orai111111”,

“modalMessage”:”The first deposit amount must be larger than 10 xrpl”,

“withdrawalMaxLimit”:30, //10% of bet amount,

“depositMinLimit”:10xrpl,

“pointConvertRatio”:{

“sol”:0.5,

“USDC”:1

},

“pointConvertLimit”:{ //1% of bet amount

“sol”:1,

“USDC”:100

}

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. GET Request(Initial login)
2. Response code

|  |  |  |
| --- | --- | --- |
| response code | response message | explanation |
| 0 | SUCCESS | success |
| 1 | INTERNAL\_ERROR | Server internal error |
| 2 | INVALID\_ACTION | request error |
| 3 | INVALID\_AGENT | agent error |
| 4 | BLOCK\_AGENT | blocked agent |
| 5 | INVALID\_USER | user error |
| 6 | BLOCK\_USER | blocked user |
| 7 | DUPLICATE\_USER | duplicate users |
| 8 | INSUFFICIENT\_MONEY | Insufficient money error |
| 12 | INVALID\_VENDOR | vendor error |
| 13 | INVALID\_PARAMETER | Request parameter error |
| 14 | NETWORK\_ERROR | network error |
| 15 | MAINTENANCE | Under maintenance |
| 18 | INVALID\_WAGER | Transaction details ID |
| 20 | INVALID\_TIME | time error |
| 21 | DUPLICATE\_REQUESTKEY | Duplicate prevention request key error |
| 22 | TIMEOUT\_ERROR | Timeout error |

1. Game type

|  |  |
| --- | --- |
| code | explanation |
| 1 | **Slot** |
| 2 | **Live Casino** |

1. Language

|  |  |
| --- | --- |
| code | explanation |
| ko | **Korean** |
| en | **English** |
| jp | **Japanese** |
| th | **Thai** |
| pt | **Portuguese** |
| mn | **Mongolian** |
| zh | **Chinese** |
| vi | **Vietnamese** |

1. Model
2. User

|  |  |  |
| --- | --- | --- |
| field | category | explanation |
| userCode | String | user code |
| balance | Decimal | User's money |