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. 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

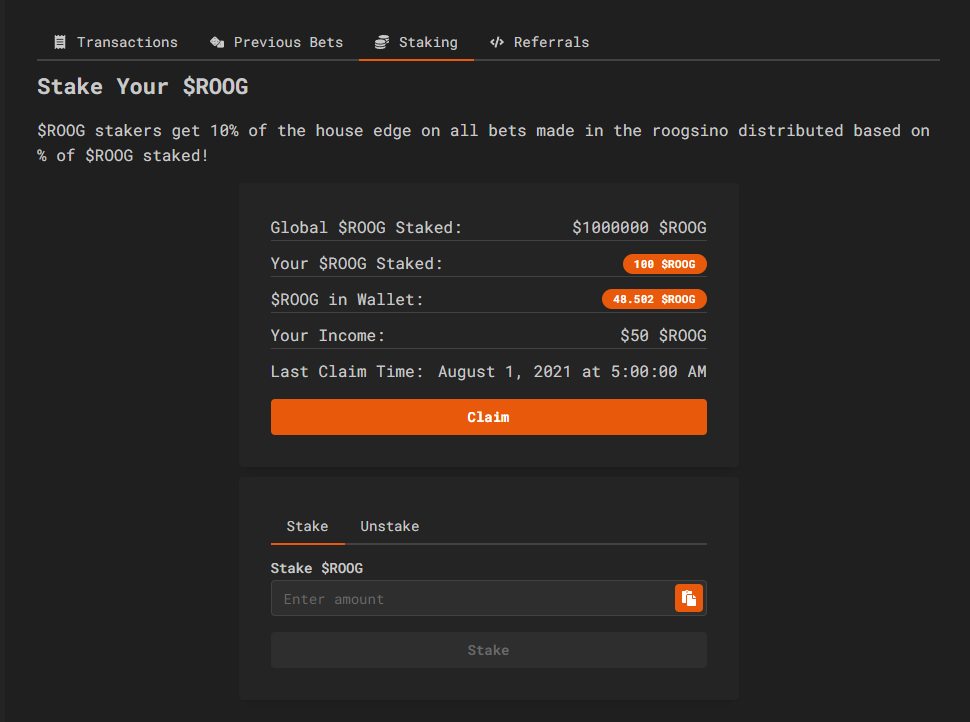
{

“status”: 1,

" msg ” : "Fail"

}

1. GetStakingInfo



This method returns the amount of $token a user has staked, the total amount of $token staked and the amount of earnings the user has.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Category | Explanation | essential |
| Method | String | GetStakingInfo | Y |
| coinType | String | Coin type | Y |

* Request example:

{

“ method” : “ GetStakingInfo”,

“ coinType” : “xrpl”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| Field | Category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| stakedAmount | Decimal | User's staking money |
| totalStakedAmount | Decimal | Total staking money |
| income | Decimal | Profit |
| lastSettleTime | DateTime |  |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

"stakedAmount" : 10000,

“totalStakedAmount”:1000000,

“income”:30, //10% of bet amount,

“lastSettleTime”: ”2024-08-29 10:00:12”

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. Stake / Unstake

This method stakes $token for profit share, the profit share should be 10% of house edge on all bets devided by the % of each persons stake.

If amount > 0, stake

If amount < 0, unstake

After API call, increase stake amount, decrease balance.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Category | Explanation | essential |
| method | String | Stake | Y |
| coinType | String | Coin type | Y |
| amount | Decimal | amount |  |

* Request example:

{

“method” : “ Stake”,

“coinType” : “xrpl”,

“amount”:1000

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| Field | Category | explanation |
| Status | Integer | Response code (see Appendix 4.1) |
| Msg | String | response message |
| balance | Decimal | User's balance |
| stakedAmount | Decimal | User’s staking money |
| income | Decimal |  |
| lastSettleTime | DateTime |  |

* Response example:
* success

{

"status" : 0,

"msg" : "SUCCESS",

"balance" : 10000,

“stakedAmount”:1000000,

“income”:30, //10% of bet amount,

“lastSettleTime”: ”2024-08-29 10:00:12”

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

1. ClaimStakingBonus

This method returns the amount of $token a user has staked, the total amount of token staked and the amount of earnings the user has.

* Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Category | Explanation | essential |
| method | String | ClaimStakingBonus | Y |
| coinType | String | Coin type | Y |

* Request example:

{

“method” : “ ClaimStakingBonus”,

“coinType” : “xrpl”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| Field | Category | explanation |
| Status | Integer | Response code (see Appendix 4.1) |
| Msg | String | response message |
| balance | Decimal | User's balance |
| stakedAmount | Decimal | User’s staking money |
| income | Decimal |  |
| lastSettleTime | DateTime |  |

* Response example:
* success

{

"status" : 0,

"msg" : "SUCCESS",

"balance" : 10000,

“stakedAmount”:1000000,

“income”:30, //10% of bet amount,

“lastSettleTime”: ”2024-08-29 10:00:12”

}

* error

{

“status”: 5,

" msg ” : "INVALID\_USER"

}

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 |

* 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

}

* 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 | GetVritualBalanceModalInfo | Y |
| coinType | String | Coin type | Y |

* Request example:

{

“ method” : “ GetVirtualBalanceModalInfo”,

“ coinType” : “USO”

}

* Response parameters

|  |  |  |
| --- | --- | --- |
| field | Category | explanation |
| status | Integer | Response code (see Appendix 4.1) |
| msg | String | response message |
| balance | Decimal | User's money |
| modalMessage | String | Message shown in modal |
| depositConvertRatio | Dictionary | If coinType is virtual |
| depositConvertMinLimit | Dictionary | If coinType is virtual |
| withdrawConvertRatio | Dictionary |  |
| withdrawConvertMaxLimit | Dictionary |  |

* Response example:
* success

{

"status" : 0,

" msg " : "SUCCESS",

"balance" : 10000,

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

“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 |