

[RREWARD API] RestAPI 요청 문서

Introduction

파이널 체인의 보상 시스템을 관리하기 위한 API 서버 입니다.

등록된 Node의 생성된 지갑의 주소들로 Transaction 정보 확인 및 자산 입금을 API 요청을 통하여 할 수 있습니다. **(진행에 따라 내용이 조금씩 수정될 수 있습니다.)**

API - Secret Key 쌍은 하나의 심볼에만 사용됩니다. (보안을 위하여 꼭 필요합니다.)

baseurl 정보는 최종 서버에 맞추어 IP 주소가 아닌 DNS 주소로 변경되거나 최종 서버의 IP로 변경됩니다.

General Info

General API Information

- API 요청 baseurl : http://ipaddress:14501/
- 테스트 API 요청 주소
 - DGOS : ????.????.???.
 - WCL : 54.164.161.105
- 모든 요청에 대한 응답은 JSON 객체로 응답
- API 버전 : V1.0.0
- Content-type : application/json

HTTP Return Codes

- HTTP 200 return : 정상 요청
- HTTP 400 return : 클라이언트 요청 초과 오류
- HTTP 429 return : 요청 수 초과 오류
- HTTP 500 return : 서버 내부 오류

General Parameter Information

이름	설명	필수 여부	타입
API Key	제공된 API Key	필수	String
Timestamp	UTC Timestamp	필수	String
Version	요청하는 API 버전	필수	String
signature	Signature data	필수	String
symbol	코인 심볼	필수	String

General Parameter 의 signature 구성

HmacSHA256 알고리즘을 이용하여 Parameter 정보를 암호화 데이터 입니다.

API Key, Timestamp, version 정보를 이용하여 Signature 데이터를 생성합니다.

Timestamp 는 꼭 현재 시간 기준으로 적용이 되어야 합니다.

아래의 apiKey 는 샘플 key 로 사용이 불가합니다.

```
String plain = "apiKey=a2310fb0bb92477b93d0608e2507955e&timestamp=2022-07-10  
14:02:56.479&version=V1.0.0"
```

```
String signature = HmacSHA256.encrypt(plain, secret_key);
```

Signature Java Example

```
public String GenerateSignature() {  
    HmacEncrypt hmacenc = new HmacEncrypt();  
    Timestamp tmStamp = Timestamp.valueOf(LocalDateTime.now());  
    String tsstr = "&timestamp=" + tmStamp;  
    String requeststr = "apiKey=a2310fb0bb92477b93d0608e2507955e" + tsstr + "&version=V1.0.0";  
    String signature = hmacenc.hget(requeststr, "38df3a8e044d47e891a51f539633a392");  
    return signature;  
}
```

Signature C# Example

```
public static string GenerateSignature(string data, string signatureKey){  
    var keyByte = Encoding.UTF8.GetBytes(signatureKey);  
    using (var hmacsha256 = new HMACSHA256(keyByte)){  
        hmacsha256.ComputeHash(Encoding.UTF8.GetBytes(data));  
        return hmacsha256.Hash.Aggregate("", (current, t) => current + t.ToString("X2")).ToLower();  
    }  
}
```

Signature PHP Example

```
<?php  
function GenerateSignature($timestamp) {  
    $requeststr = "apiKey=a2310fb0bb92478a93d0608e2507955e&timestamp=".$timestamp."&version=V1.0.0";  
    $signature = hash_hmac('sha256', $requeststr, '38df3a8e044d47e891a51f539633a392');  
    return $signature;  
}  
  
$sig = GenerateSignature("2022-07-12 04:22:01.000");  
echo $sig;
```

HMAC SHA256 ?

일반 SHA256 과, HMAC SHA256 은 해싱 방식이 조금 다른데요, HMAC SHA256 의 경우엔 데이터를 주어진 비밀키(secret) 와 함께 해싱을 하고, 해싱된 결과물을 비밀키와 함께 다시 한번 해싱을 합니다.

이렇게 함으로서, 해시의 보안이 강화됩니다.

SIGNED Endpoint security

API 서버에 정보를 요청하기 위해서는 요청 바디에 HMAC_SHA256 암호화된 Signature 데이터를 추가해야 합니다.

API Key와 Parameter들을 Secret Key로 HMAC_SHA256으로 암호화합니다.

요청 시 timestamp 값을 바디에 추가하는데, 이는 서버에서 실제 받은 시간과 비교해 요청을 받아드립니다. (default 1분)

Secret Key의 관리는 매우 중요합니다. 이 key 값은 철저히 관리해주세요.

이 key의 유출로 인하여 발생한 Transaction은 정상적인 요청으로 밖에 관리될 수 없습니다.

Transaction 생성의 악의적인 공격은 Secret Key 유출로부터 발생됨을 꼭 명심하세요.

이러한 **Secret Key의 유출로 인한 피해는 본 시스템에는 아무런 책임이 없음**을 명시합니다.

만약 key가 유출되었다고 의심이 되거나 이상한 Transaction이 발생되었다면 BITSTOA 관리자의 연락을 통하여 Secret Key를 재설정 해주시기 바랍니다.

Error Codes

10001 **INVALID_TIMESTAMP**

요청 Timestamp이 유효하지 않습니다.

요청 Timestamp는 서버 시간과 1분 이내일 때 요청을 수락합니다.

```
{"code": 10001,"status": "error","message": "Timestamp for this request was 1000ms ahead of the server's time."}
```

10002 **BAD_SYMBOL**

API 서버가 지원하지 않는 심볼입니다.

```
{"code": 10002,"status": "error","message": "Invalid symbol"}
```

10003 **INVALID_SIGNATURE**

요청된 Signature 값이 유효하지 않습니다.

```
{"code": 10003,"status": "error","message": "Signature for this request is not valid."}
```

10005 **TOO_MANY_REQUESTS**

API 서버에 현재 처리중인 큐가 가득 찼습니다.

일정 시간 뒤에 정상적 요청 가능합니다.

```
{"code": 10005,"status": "error","message": "Too many requests queued."}
```

10006 **INVALID_PARAM**

요청 Parameter 값이 유효하지 않습니다.

필수 Parameter 값이 존재하지 않습니다.

```
{"code": 10006,"status": "error","message": "Invalid data sent for a parameter."}
```

10007 **INNER_FAILURE**

내부 서버 에러입니다.

```
{"code": 10007,"status": "error","message": "Internal server error."}
```

10008 **Deposit error**

입금 관련 에러입니다.

거래엔진 연결 문제, 입금 상태에 따라 발생합니다.

```
{"code": 10008,"status": "error","message": "Error Message."}
```

10009 **IP Error**

Private API 요청 시에 사용되는 IP와 일치하지 않을 때 발생하는 에러입니다.

Node 등록 요청

- http://{baseurl}/v1/node/insert POST

지정된 주소로부터 요청된 주소로 Node의 등록 정보를 전송합니다.

Node의 등록과 지갑의 등록이 같이 진행됩니다.

node 정보는 GPI 쪽의 symbol에 "node"를 입력해주세요.

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
serial	Node 의 고유한 번호	필수	String
ipa	node 의 IP Address	선택	String
price	node 의 단가	필수	String
maxp	Node 보상의 최대 크기	필수	String
reward	Node 에 보상된 금액	필수(초기 : 0)	String
startday	보상이 시작되는 시간	필수	String
lastday	보상이 만료되는 시간	필수	String
op_time	가동된 시간 정보	필수	String
addr	Node 의 주소	필수	String

Request

```
{"apiKey":"a2310fb0bb92477b93d0608e2507955e","timestamp" : "2022-07-10 13:32:55.818","version" : "V1.0.0","signature" : "817b4818b7bdf41a10b4c46320fedae711cca714087bb36351398bb681c29579","symbol":"node","serial": "sn12345678","ipa":"192.168.100.1","price":"1000000","maxp":"200000","reward":"0","startday":"20220801","lastday":"20320831","op_time":"2022-08-01 00:00:00","addr" : "0xca23fAA3936Cf211b83EeC2eC4eb408dD63d02fC"}
```

Response Parameters (data 영역)

이름	설명	필수 여부	타입
v_amount	지갑의 Balance	필수	String
t_amount	사용 예정 금액	필수	String
c_tm	입금된 주소	필수	String
addr	등록된 주소	필수	String

Response

```
{"msg":"success","data":{"v_amount":"0.000000000000000000","t_amount":"0.000000000000000000","addr":"0xca23fAA3936Cf211b83EeC2eC4eb408dD63d02fC","c_tm":"2022-08-01 00:00:00"},"status":true}
```

Node 보상 입금 요청

- `http://{baseurl}/v1/node/transfer` POST

지정된 주소로부터 요청된 주소로 Node 보상의 수행을 전송합니다.

node_wallet에 포함된 froma 주소로부터 toa 주소로 지정된 amount 만큼의 자산을 이동합니다.

froma는 관리를 위해 지정된 node들의 전체 보상을 관리하는 지갑입니다. 모든 지갑은 이 지갑에 충전된 코인의 한도에서 자산이 전송됩니다.

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
froma	소스 주소 지정	필수	String
toa	목적 주소 지정	필수	String
amount	보상을 전송할 양	필수	String

Request
{
 "apiKey": "a2310fb0bb92477b93d0608e2507955e",
 "timestamp": "2022-07-10 13:32:55.818",
 "version": "V1.0.0",
 "signature": "817b4818b7bdf41a10b4c46320fedae711cca714087bb36351398bb681c29579",
 "symbol": "node",
 "froma": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2",
 "toa": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3",
 "amount": "100.000000000000000000"
}

Response Parameters (data 영역)

이름	설명	필수 여부	타입
v_amount	지갑의 Balance	필수	String
t_amount	사용 예정 금액	필수	String
c_tm	입금된 주소	필수	String
addr	등록된 주소	필수	String

Response (toa의 지갑 정보로 응답)
{
 "msg": "success",
 "data": {
 "v_amount": "100.000000000000000000",
 "t_amount": "0.000000000000000000",
 "addr": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3",
 "c_tm": "2022-08-01 00:00:00",
 "status": true
 }
}

Node 지갑 정보 요청

- http://{baseurl}/v1/node/select POST

지정된 addr 정보를 기반으로 node_wallet의 지갑의 정보를 가져옵니다..

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
addr	검색할 Node 의 주소 정보	필수	String

Request
{
 "apiKey": "a2310fb0bb92477b93d0608e2507955e",
 "timestamp": "2022-07-10 13:32:55.818",
 "version": "V1.0.0",
 "signature": "817b4818b7bdf41a10b4c46320fedae711cca714087bb36351398bb681c29579",
 "symbol": "node",
 "addr": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2"
}

Response Parameters (data 영역)

이름	설명	필수 여부	타입
v_amount	지갑의 Balance	필수	String
t_amount	사용 예정 금액	필수	String
c_tm	입금된 주소	필수	String
addr	등록된 주소	필수	String

Response
{
 "msg": "success",
 "data": {
 "v_amount": "100.000000000000000000",
 "t_amount": "0.000000000000000000",
 "addr": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3",
 "c_tm": "2022-08-01 00:00:00",
 "status": true
 }
}

User 지갑 정보 요청

- http://{baseurl}/v1/user/select POST

지정된 addr 정보를 기반으로 user_wallet의 지갑의 정보를 가져옵니다..

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
addr	검색할 Node 의 주소 정보	필수	String

Request
{
 "apiKey": "a2310fb0bb92477b93d0608e2507955e",
 "timestamp": "2022-07-10 13:32:55.818",
 "version": "V1.0.0",
 "signature": "817b4818b7bdf41a10b4c46320fedae711cca714087bb36351398bb681c29579",
 "symbol": "user",
 "addr": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3"
}

Response Parameters (data 영역)

이름	설명	필수 여부	타입
v_amount	지갑의 Balance	필수	String
t_amount	사용 예정 금액	필수	String
c_tm	입금된 주소	필수	String
addr	등록된 주소	필수	String

Response
{
 "msg": "success",
 "data": {
 "v_amount": "100.000000000000000000",
 "t_amount": "0.000000000000000000",
 "addr": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3",
 "c_tm": "2022-08-01 00:00:00",
 "status": true
 }
}

User 지갑 추가 요청

- http://{baseurl}/v1/user/insert POST

지정된 addr 정보를 기반으로 user_wallet에 지갑을 추가합니다..

uid 정보를 같이 전달하여 사용자 정보도 같이 등록하게 되며 등록된 user_no를 기반으로 전달된 사용자의 지갑 address를 저장한다.

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
addr	검색할 Node의 주소 정보	필수	String
uid	사용자를 구분하기 위한 유일한 ID(email 등)	필수	String
Request { "apiKey": "a2310fb0bb92477b93d0608e2507955e", "timestamp": "2022-07-10 13:32:55.818", "version": "V1.0.0", "signature": "817b4818b7bdf41a10b4c46320fedae711cca714087bb36351398bb681c29579", "symbol": "user", "addr": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB4", "uid": "test@test.com" }			

Response Parameters (data 영역)

이름	설명	필수 여부	타입
v_amount	지갑의 Balance	필수	String
t_amount	사용 예정 금액	필수	String
c_tm	입금된 주소	필수	String
addr	등록된 주소	필수	String
Response { "msg": "success", "data": { "v_amount": "0.000000000000000000", "t_amount": "0.000000000000000000", "addr": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB4", "c_tm": "2022-08-01 00:00:00", "status": true } }			

User 구매 이력 등록

- http://{baseurl}/v1/user/buyadd POST

node의 보상을 분배하기 위한 노드별 구매 이력을 tbl_buy_history 테이블로 등록한다.

tbl_buy_history에 등록된 내용에 따라 node_wallet에 입금된 보상 금액이 각 node 구매자들에게 분할되어 입금 처리됩니다. (node_wallet -> user_wallet)

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
nodesn	Node 의 고유한 번호	필수	String
price	node 의 단가	필수	String
maxp	Node 보상의 최대 크기	필수	String
reward	Node 에 보상된 금액	필수(초기 : 0)	String
startday	보상이 시작되는 시간	필수	String
lastday	보상이 만료되는 시간	필수	String
addr	구매 회원의 주소	필수	String
naddr	Node 의 주소	필수	String

Request

```
{ "apiKey": "a2310fb0bb92477b93d0608e2507955e", "timestamp": "2022-07-10 13:32:55.818", "version": "V1.0.0", "signature": "817b4818b7bdf41a10b4c46320fedae711cca714087bb36351398bb681c29579", "symbol": "user", "nodesn": "sn12345678", "price": "100000", "maxp": "20000", "reward": "0", "startday": "20220801", "lastday": "20320831", "addr": "0xaa23fAA3936Cf211b83EeC2eC4eb408dD63d02fC", "naddr": "0xca23fAA3936Cf211b83EeC2eC4eb408dD63d02fC" }
```

Response Parameters (data 영역)

이름	설명	필수 여부	타입
v_amount	지갑의 Balance	필수	String
t_amount	사용 예정 금액	필수	String
c_tm	입금된 주소	필수	String
addr	등록된 주소	필수	String

Response (구매 요청에 대한 내용을 그대로 리턴함)

```
{ "msg": "success", "data": { "nodesn": "sn12345678", "price": "100000", "maxp": "20000", "reward": "0", "startday": "20220801", "lastday": "20320831", "addr": "0xaa23fAA3936Cf211b83EeC2eC4eb408dD63d02fC", "naddr": "0xca23fAA3936Cf211b83EeC2eC4eb408dD63d02fC"}, "status": true }
```

User 지갑에서의 unlock 처리 요청

- http://{baseurl}/v1/user/unlockwallet POST

lock 되어 있는 tbl_in의 내용들을 검색하여 unlock을 수행합니다.

email 정보를 같이 전달하여 email을 통하여 사용자 정보의 user_no를 찾을 수 있어야 한다.

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
FROMD	Unlock 처리를 위한 검색 시작시간	필수	String
TOD	Unlock 처리를 위한 검색 마지막 시간	필수	String
Request { "apiKey": "a2310fb0bb92477b93d0608e2507955e", "timestamp": "2022-07-10 13:32:55.818", "version": "V1.0.0", "signature": "817b4818b7bdf41a10b4c46320fedae711cca714087bb36351398bb681c29579", "symbol": "user", "FROMD": "2022-07-10 00:00:00", "TOD": "2022-07-10 23:59:59" }			

Response Parameters (data 영역)

이름	설명	필수 여부	타입
v_amount	지갑의 Balance	필수	String
t_amount	사용 예정 금액	필수	String
c_tm	입금된 주소	필수	String
addr	등록된 주소	필수	String
Response { "msg": "success", "data": { "v_amount": "0.000000000000000000", "t_amount": "0.000000000000000000", "count": 10 }, "status": true }			

count 로 unlock 처리한 건수의 숫자를 응답합니다.

사용자 정보 등록 요청

- `http://{baseurl}/v1/user/useradd` POST

회원의 정보의 입력을 요청합니다..

email 정보를 꼭 추가해야 하며 email을 기준으로 사용자 정보의 user_no 할당합니다.

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

-> 이 부분이 아직 어떻게 정보를 받아서 관리하게 될지 정해지지 않음

View를 위한 지갑 정보 요청

- http://{baseurl}/v1/user/select POST

지정된 addr 정보를 기반으로 \$(symbol)_wallet의 지갑의 정보를 가져옵니다.

사용자 정보는 GPI 쪽의 symbol에 "node" 또는 "user"를 입력해주세요.

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
테이블들의 조건	검색할 조건	선택	String
Request { "apiKey": "a2310fb0bb92477b93d0608e2507955e", "timestamp": "2022-07-10 13:32:55.818", "version": "V1.0.0", "signature": "817b4818b7bdf41a10b4c46320fedae711cca714087bb36351398bb681c29579", "symbol": "user", "addr": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2" }			

Response Parameters (data 영역)

이름	설명	필수 여부	타입
v_amount	지갑의 Balance	필수	String
t_amount	사용 예정 금액	필수	String
c_tm	입금된 주소	필수	String
addr	등록된 주소	필수	String
Response { "status": "true", "message": "Success", "data": { "contents": { "no": 1, "amount": "0.000000000000000000", "weight": 100, "taker_fee": 0.040, "i_amount": "0.000000000000000000", "t_amount": "0.000000000000000000", "t_cnt": 0, "inout": "1", "prev_amount": "0.000000000000000000", "txcount": 0, "t_fee": "0.000000000000000000", "u_tm": "2022-07-15 09:41:40", "v_amount": "1000000.000000000000000000", "cma": "0", "maker_fee": 0.040, "c_tm": "2022-07-14 20:02:17", "addr": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2", "usr_no": "1" } } }			

data -> contents 에 데이터 정보가 포함됨

View를 위한 지갑 리스트 정보 요청

- http://{baseurl}/view/wallet/selectlist POST

지갑 테이블의 검색 조건에 따라 \$(symbol)_wallet 데이터 베이스의 tbl_wallet의 지갑 정보들을 가져옵니다.

검색 조건들은 테이블의 컬럼들을 조건으로 추가할 수 있습니다.

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
addr	검색할 Node 의 주소 정보	필수	String

Request
{ "apiKey": "a2310fb0bb92478a93d0608e2507955f", "timestamp": "2022-07-18 13:57:43.731", "version": "V1.0.0", "signature": "b6dfefee656e85a957a20205c120e71bbc7efd248f78d45c1502b0cb61024b29", "symbol": "user" }

Response Parameters (data 영역)

Response
{ "apiKey": "a2310fb0bb92478a93d0608e2507955f", "timestamp": "2022-07-18 13:57:43.731", "version": "V1.0.0", "signature": "b6dfefee656e85a957a20205c120e71bbc7efd248f78d45c1502b0cb61024b29", "symbol": "user" }

RESPONSE :
{"status": "true", "message": "Success", "data": { "reqList": [{ "no": 2, "amount": "0.000000000000000000", "weight": 100, "taker_fee": 0.040, "i_amount": "0.000000000000000000", "t_amount": "300.000000000000000000", "t_cnt": 0, "inout": "1", "prev_amount": "0.000000000000000000", "txcount": 0, "t_fee": "0.000000000000000000", "u_tm": "2022-07-15 09:41:40", "v_amount": "400.000000000000000000", "cma": "0", "maker_fee": 0.040, "c_tm": "2022-07-14 20:02:17", "addr": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3", "usr_no": "2" }, { "no": 1, "amount": "0.000000000000000000", "weight": 100, "taker_fee": 0.040, "i_amount": "0.000000000000000000", "t_amount": "0.000000000000000000", "t_cnt": 0, "inout": "1", "prev_amount": "0.000000000000000000", "txcount": 0, "t_fee": "0.000000000000000000", "u_tm": "2022-07-15 09:41:40", "v_amount": "999600.000000000000000000", "cma": "0", "maker_fee": 0.040, "c_tm": "2022-07-14 20:02:17", "addr": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2", "usr_no": "1" }], "rstatus": 1, "totalCount": 2 } }

data -> contents 에 데이터 정보가 포함됨

View를 위한 입금/출금 테이블 정보 요청

- `http://{baseurl}/view/trxin/select` POST
- `http://{baseurl}/view/trxout/select` POST

지갑 테이블의 검색 조건에 따라 \$(symbol)_wallet 데이터 베이스의 tbl_in, tbl_out의 거래 정보 1건을 가져옵니다.

검색 조건들은 테이블의 컬럼들을 조건으로 추가할 수 있습니다.

General Parameter Information에 다음의 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
테이블들의 조건	검색할 조건	선택	String
Request { "apiKey": "a2310fb0bb92478a93d0608e2507955f", "timestamp": "2022-07-18 14:32:05.599", "version": "V1.0.0", "signature": "be986a9b8d23dd5e9c6fb8eb61f9a0cbe08e81e4557804d8f18e2e31113a82d7", "symbol": "user", "FROMD": "2022-07-16 00:00:00", "TOD": "2022-07-17 23:59:59" }			

Response Parameters (data 영역)

Response { "status": "true", "message": "Success", "data": { "contents": { "no": 1, "amount": "100", "txid": "dXNya0xMYiDUxKOCX5qbj1657983385225", "from": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2", "c_tm": "2022-07-16 23:56:25", "to": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3", "confirm_num": "7", "usr_no": "2", "email": "roy@hackersholidays.com" } } }

data -> eqList 에 지갑들의 데이터 정보가 포함됨

data -> totalCount 에 데이터 리스트의 개수 정보가 포함됨

입금/출금 정보 요청

- http://{baseurl}/view/trxin/selectlist POST
- http://{baseurl}/view/trxout/selectlist POST

지갑 테이블의 검색 조건에 따라 \$(symbol)_wallet 데이터 베이스의 tbl_in, tbl_out의 거래 정보들을 가져옵니다.

검색 조건들은 테이블의 컬럼들을 조건으로 추가할 수 있습니다.

General Parameter Information에 추가적인 Parameter들이 추가됩니다.

추가 Request Parameter

이름	설명	필수 여부	타입
addr	검색할 Node 의 주소 정보	필수	String
Request { "apiKey": "a2310fb0bb92478a93d0608e2507955f", "timestamp": "2022-07-18 14:37:26.106", "version": "V1.0.0", "signature": "d878258370e774df500839103e9fe51a699c8398723fd983f4a506174b85cf79", "symbol": "user", "FROMD": "2022-07-16 00:00:00", "TOD": "2022-07-17 23:59:59" }			

Response Parameters (data 영역)

Response { "status": "true", "message": "Success", "data": { "reqList": [{ "no": 4, "amount": "100", "txid": "dXNya0u6m0ssozsP68s711657983670390", "from": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2", "c_tm": "2022-07-17 00:01:10", "to": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3", "confirm_num": "9", "usr_no": "2" }, { "no": 3, "amount": "100", "txid": "dXNya0L1i5y3o7qjWHffA1657983424488", "from": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2", "c_tm": "2022-07-16 23:57:04", "to": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3", "confirm_num": "9", "usr_no": "2" }, { "no": 2, "amount": "100", "txid": "dXNya0eEp5cEFdqTnif981657983385235", "from": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2", "c_tm": "2022-07-16 23:56:25", "to": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3", "confirm_num": "9", "usr_no": "2" }, { "no": 1, "amount": "100", "txid": "dXNya0xMYiDUxKOCX5qbj1657983385225", "from": "0xF1FD14cBB10ab7EC912E6617C1b6E74a50B39bB2", "c_tm": "2022-07-16 23:56:25", "to": "0xA1FD14cBB10ab7EC912E6617C1b6E74a50B39bB3", "confirm_num": "7", "usr_no": "2" }], "rstatus": "1", "totalCount": 4 } }			
---	--	--	--

data -> contents 에 데이터 정보가 포함됨

Node 및 구매 내역에 따른 정보 요청

- http://{baseurl}/view/buyhist/selectlist POST
- http://{baseurl}/view/buyhist/selectlist POST
- http://{baseurl}/view/nodehist/selectlist POST
- http://{baseurl}/view/nodehist/selectlist POST
- http://{baseurl}/view/unlockhist/selectlist POST
- http://{baseurl}/view/unlockhist/selectlist POST

Node와 구매한 내역, 그리고 Lock 해제된 정보들을 가져옵니다.

tbl_node_history의 정보를 획득합니다.

이름:

코멘트:

열:

추가

제거

위로

아래로

#	이름	데이터 유형	코멘트	00
1	idx	INT	Node의 index	
2	serial	VARCHAR	Node의 고유 번호	
3	ipa	VARCHAR	Node에 할당된 ip address	
4	price	VARCHAR	Node의 구매 금액	
5	maxp	VARCHAR	Node 보상의 최대 크기 지급시 이 금액을 넘...	
6	reward	VARCHAR	Node 보상의 최대 크기 지급시 이 금액을 넘...	
7	startday	INT	Node가 가동되어 보상이 시작된일	
8	lastday	INT	Node의 보상이 만료되는 마지막 지급일	
9	op_time	TIMESTAMP	Node가 가동된 시간 정보	
10	c_tm	TIMESTAMP	노드 등록일	
11	u_tm	TIMESTAMP	수정일	
12	d_tm	TIMESTAMP	삭제일	

Node들의 지갑 정보와 보상 기본 정보를 관리합니다.

tbl_buy_history의 정보를 획득합니다.




열:

추가

제거

위로

아래로

#	이름	데이터 유형	코멘트
	1 idx	INT	history의 index
	2 nidx	INT	사용자가 구매에 참여한 Node의 index 번호
	3 status	INT	해당 구매 history 의 유지 상태 (0 : 미사용, ...
	4 tid	VARCHAR	TGC ID
	5 wid	VARCHAR	파이널 제인의 Wallet ID
	6 addr	VARCHAR	구매에 참여한 사용자 주소
	7 naddr	VARCHAR	노드의 지갑주소
	8 nftid	VARCHAR	사용자의 NFT ID로 메인넷 API를 통하여 획득
	9 hash	VARCHAR	사용자의 HASH KEY로 메인넷 API를 통하여 ...
	10 price	VARCHAR	사용자가 구매에 참여한 Node에 구매 참여 ...
	11 maxp	VARCHAR	사용자의 보상의 최대 크기 지급시 이 금액...
	12 reward	VARCHAR	보상된 reward (maxp에 도달하거나 lastday...
	13 startday	INT	사용자에게 보상이 시작된일
	14 lastday	INT	사용자의 보상이 만료되는 마지막 지급일
	15 c_tm	TIMESTAMP	history 등록일
	16 u_tm	TIMESTAMP	
	17 d_tm	TIMESTAMP	

회원들의 구매 내역과 보상 기준 정보를 관리합니다.

, tbl_unlock_history의 정보를 획득합니다.

#	이름	데이터 유형	코멘트
1	idx	INT	
2	usr_no	INT	보상이 지급된 사용자 번호
3	amount	VARCHAR	사용자에게 locked된 amount
4	txid	VARCHAR	사용자에 입금된 tbl_in의 txid
5	c_tm	TIMESTAMP	보상이 지급된 시기

회원들의 Lock이 풀린 보상 정보들을 관리합니다.

검색 조건들은 위 그림 테이블들의 컬럼들을 조건으로 추가할 수 있습니다.

General Parameter Information에 추가적인 Parameter들이 추가됩니다.