

修订历史

版本	日期	修订人	说明
1.0	2020-11-12	戴群彪	

火链科技区块链积分平台接口文档

目录

1	概述	6
1.1	说明	6

1.2	接口方案	6
1.2.1.	接口规范.....	6
1.2.2.	URL 组成	6
1.2.3.	HTTP 请求方式	6
1.2.4.	输入输出参数.....	6
2	接口	7
2.1.	获取 API 令牌	7
2.1.1	参数	7
2.1.2	说明	8
2.2.	添加账户	8
2.2.1	参数	8
2.2.2	说明	9
2.3.	交易上链	9
2.3.1	参数	10
2.3.2	说明	10
2.4.	根据交易 ID 获取记录.....	10
2.4.1	参数	11
2.4.2	说明	11
2.5.	查询组织加入的通道.....	11
2.5.1	参数	11
2.5.2	说明	12
2.6.	创建通道.....	12
2.6.1	参数	12
2.6.2	说明	13
2.7.	加入通道.....	13
2.7.1	参数	13
2.7.2	说明	14
2.8.	查看已安装或部署的链码.....	14
2.8.1	参数	14
2.8.2	说明	15
2.9.	调用链码函数.....	15
2.9.1	参数	15
2.9.2		16

2.10.查询链码	16
2.10.1 参数	16
2.10.2 说明	17
2.11.查看区块列表	17
2.11.1 参数	17
2.11.2 说明	18
2.12.查看交易列表	18
2.12.1 参数	19
2.12.2 说明	20
2.13.删除账户	20
2.13.1 参数	20
2.13.2 说明	21
2.14.查看通道区块高度	21
2.14.1 参数	21
2.14.2 说明	22
2.15.查看通道的 Peer 节点数	22
2.15.1 参数	22
2.15.2 说明	23
2.16.查看通道的 order 节点数	23
2.16.1 参数	23
2.16.2 说明	24
2.17.查看通道的链码数	24
2.17.1 参数	24
2.17.2 说明	25
2.18.查看通道信息	25
2.18.1 参数	26
2.18.2 说明	26
2.19.查看通道的 Peer 节点数	26
2.19.1 参数	27
2.19.2 说明	27
2.20.查看区块信息通过 Hash	27
2.20.1 参数	28
2.20.2 说明	28

2.21.查看区块头信息	29
2.21.1 参数	29
2.21.2 说明	29
2.22.通过区块号获取区块信息.....	29
2.22.1 参数	30
2.22.2 说明	30
2.23.安装链码	31
2.23.1 参数	31
2.24.通过交易 ID 获取历史数据	32
2.24.1 参数	32
2.24.2 说明	33
2.25.通过 owner 获取交易数据.....	33
2.25.1 参数	33
2.25.2 说明	34
2.26.获取最近的交易列表	34
2.26.1 参数	35
2.26.2 说明	36
2.27.通过交易 ID 获取交易信息	36
2.27.1 参数	36
2.27.2 说明	37
2.28.获取网络的所有节点	37
2.28.1 参数	37
2.28.2 说明	38
2.29.获取网络所有的智能合约.....	38
2.29.1 参数	38
2.29.2 说明	39
3 积分商城 Demo 智能合约 V1 接口	40
3.1.创建账户.....	40
3.1.1 参数	40
3.1.2 说明	41
3.2.积分发行.....	41
3.2.1 参数	41

3.2.2 说明	42
3.3.积分增发.....	42
3.3.1 参数	42
3.3.2 说明	43
3.4.积分销毁.....	43
3.4.1 参数	43
3.4.2 说明	44
3.5.积分通兑.....	44
3.5.1 参数	44
3.5.2 说明	45
3.6.账户单币种余额查询.....	45
3.6.1 参数	46
3.6.2 说明	46
3.7.账户余额查询.....	47
3.7.1 参数	47
3.7.2 说明	48
3.8.积分详情查询.....	48
3.8.1 参数	48
3.8.2 说明	49
3.9.交易记录查询.....	49
3.9.1 参数	49
3.9.2 说明	50
3.10.所有兑换交易记录查询.....	50
3.10.1 参数	50
3.10.2 说明	51
3.11.某账户发行的所有积分情况.....	51
3.11.1 参数	51
3.11.2 说明	52

1 概述

1.1 说明

本文档主要针对 fabric-sdk-node 的调用参考

阅读对象：开发人员。

1.2 接口方案

1.2.1. 接口规范

1.2.2. URL 组成

1.2.3. HTTP 请求方式

1.2.4. 输入输出参数

2 接口

2.1. 获取 API 令牌

通过该接口，用户可以获取接口调用所需的令牌，其他的接口调用都需要该令牌。请求时该令牌通过 Header 来传递。

```
header: {  
  'Accept': 'application/json',  
  'Content-Type': 'application/json',  
  'x-access-token': localStorage.getItem('token')  
}
```

接口地址： Get 方法/:alliance/:channel/users/getUserLoginToken

例子：

<http://apibaas.hnsoft.com/alliance/channel/users/getUserLoginToken>

参数：

```
req.userName  
req.orgName
```

2.1.1 参数

➤ body 参数

参数	参数名称	参数说明	样例
userName	用户账号		
orgName	用户组织		

➤ 响应

成功返回信息示例如下：

```
1 {  
2   "success": true,  
3   "message": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOiJlbnZyNTk4OTAsInVzZXQyZW1lIjo1YWwtaw41LCJvcnd0YV1lIjoib3JhNSIsInldCI6MTU3NjcyZg5MH0. bR4RswRvSgpc9rem4MFH9ATjzvt2nV_GLSTE9kvpRY"  
4 }
```

失败返回的示例如下：

```
1  {
2    "success": false,
3    "message": "This user has not exist, pls register user first!"
4  }
```

2.1.2 说明

获取用户 token 信息

2.2. 添加账户

接口地址：POST 方法 /:alliance/:channel/:org/user/register

例子：<http://apibaas.hnsoft.com/dev-cybernet01/mychannel/org1/user/register>

2.2.1 参数

➤ url 参数

: alliance	联盟名称	
: channel	通道名称	
: org	组织名称	

➤ body 参数

参数	参数名称	参数说明	样例
userName	用户名	账户登录的唯一ID	
password	密码	10位以上符合安全要求的	
affiliation	组织单位	账户所隶属于的单位组织	

role	角色类型	账户角色	Eg. 'client'
maxEnrollments	最大可注册用户数量	默认值0，最大不超过CA配置里的设定	0
revoker	吊销角色	是否可以吊销其他账户	'true'
registrarRoles	账户类型列表	做为注册商，可注册账户类型列表	'client,user'
registrarAttributes	额外属性列表	做为注册商，可注册账户的额外属性列表	'*'

➤ 响应

用户注册成功信息返回用户私钥。

```
{
  "success": true,
  "result": {
    "secret": "QZJFemSGfBZL",
    "privateKey": "-----BEGIN PRIVATE KEY-----\r\nMIGHAgEAMBMGBYqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgMm1ZbG95a1qXGDY1\r\nnddw\r\n+TSH5PtWDgr9qiimdpYhLQHWhRANCAApHUYvXYLPwxaBGr0eKG5CfhVxz97K\r\n\r\nWJjy3KECmzJcPWVRgEYI\r\nr1zAbnU\r\n+h6g6qzceu1w38BK035503RmruiD\r\n\r\n-----END PRIVATE KEY-----\r\n"
  }
}
```

用户注册失败返回失败信息

```
1 {
2   "success": false,
3   "err": {
4     "message": "fabric-ca request register failed with errors [{"code":0,"message":\\"Registration of 'daigunbiao' failed: Identity 'daigunbiao' is already registered"}]"
  }
}
```

2.2.2 说明

2.3. 交易上链

接口地址：POST 方法 /:alliance/:channel/registerTran

Form-data 形式

例子：<http://apibaas.hnsoft.com/alliance/channel/registerTran>

参数：

Req. body. owner

Req. body. ID

Req. body. data

Req. body. org

2.3.1 参数

➤ URL 参数

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		

➤ body 参数

参数	参数名称	参数说明	样例
org	组织名称	上链用户所属单位	
owner	拥有者		
ID	交易ID		
data	交易内容		

➤ 响应

合约数据返回

2.3.2 说明

查询方法 根据合约返回的数据进行解析操作

2.4.根据交易 ID 获取记录

接口路径: GET 方法 /:alliance/:channel/querybyID/:id

例子: <http://apibaas.hnsoft.com/alliance/channel/querybyID/id>

2.4.1 参数

➤ URL 参数

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: id	交易ID		

➤ 响应

按合约数据格式返回

Pretty	Raw	Preview	Visualize BETA	HTML	↻
1 [{"Name": "15", "Amount": {"owner": "daiqunbiao", "ID": "15", "data": "0089992221111223434"}}]					

2.4.2 说明

2.5.查询组织加入的通道

接口: GET 方法 /:alliance/:organization/channels

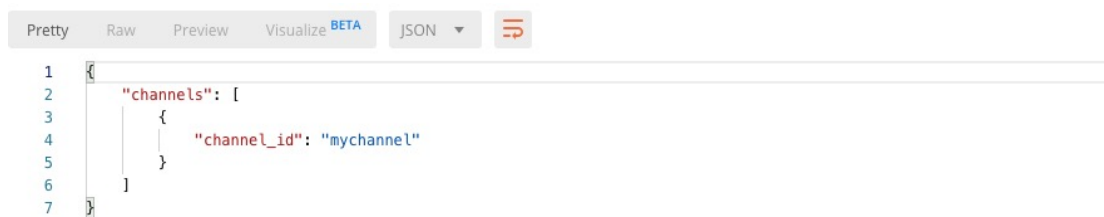
例子: <http://apibaas.hnsoft.com/alliance/org1/channels>

2.5.1 参数

➤ URL 参数

参数	参数名称	参数说明	样例
: alliance	网络名称		
: organization	组织名称		

➤ 响应



2.5.2 说明

2.6.创建通道

接口地址：/:alliance/:organization/channel

例子：http://apibaas.hnsoft.com/dev-cybernet01/org1/channel

2.6.1 参数

➤ url 参数

参数	参数名称	参数说明	样例
:alliance	联盟名称	用联盟名称替换	dev-cybernet01
:organization	组织名称名称	用组织名称替换	org1

➤ form-data 参数

参数	参数名称	参数说明	样例
channelTX	通道配置文件	参数类型：文件	
orderer	排序节点名称	目标排序节点名称	

members	成员组织名称	通道的成员组织列表	
channelName	通道名称	通道名称	

➤ 响应

```
"success": true,  
"message": "Channel 'channel-test-09' created Successfully"
```

2.6.2 说明

2.7.加入通道

接口地址: /:alliance/:organization/channel/join

例子: <http://apibaas.hnsoft.com/alliance/organization/channel/join>

2.7.1 参数

➤ url 参数

参数	参数名称	参数说明	样例
:alliance	联盟名称	用联盟名称替换	dev-cybernet01
:organization	组织名称名称	用组织名称替换	Org1

00

➤ body 请求

参数	参数名称	参数说明	样例
channelName	通道名称	要加入的通道名称	
peers	节点列表	要加入通道的节点名称	

➤ 响应

```
{  
  "success": true,  
  "message": "Successfully joined peers in Organization \"org1\" to Channel: channel-test-09"  
}
```

2.7.2 说明

返回参数

2.8.查看已安装或部署的链码

接口：GET 方法 /:alliance/:organization/:peer/chaincodes

例子：<http://apibaas.hnsoft.com/alliance/org1/peername/chaincodes>

2.8.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
:alliance	网络名称		
:organization	组织名称		
:peer	节点名称		

➤ 响应

```
Pretty Raw Preview Visualize BETA JSON ↕
1  {
2    "chaincodes": [
3      {
4        "name": "huobicc",
5        "version": "1.0",
6        "path": "github.com/chaincode/baas/go/",
7        "input": "",
8        "escc": "",
9        "vscc": "",
10       "id": {--
182     }
183   },
184   {
185     "name": "mycc",
186     "version": "1.0",
187     "path": "github.com/chaincode/sample/go/",
188     "input": "",
189     "escc": "",
190     "vscc": "",
191     "id": {
192       "buffer": {--
357     },
358     "offset": 129,
359     "markedOffset": -1,
360     "limit": 161,
361     "littleEndian": true,
362     "noAssert": false
363   }
364 }
365 ]
366 }
```

2.8.2 说明

返回参数

2.9.调用链码函数

接口地址: `/:alliance/:channel/:org/chaincode/invoke`

例子: `http://apibaas.hnsoft.com/alliance/channel/org/chaincode/invoke`

2.9.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		

: channel	通道名称		
: org	组织名称		

➤ **body 请求**

参数	参数名称	参数说明	样例
ccname	合约名称		
fcn	函数名称		
args	参数列表		

➤ **响应**

返回合约数据结果

2.9.2

返回参数

2.10.查询链码

接口地址：POST 方法 /:alliance/:channel/:org/chaincode/query

例子：http://apibaas.hnsoft.com/alliance/channel/org/chaincode/query

2.10.1 参数

➤ **URL 请求**

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ **body 请求**

参数	参数名称	参数说明	样例
ccname	链码名称		
fcn	函数名称		
args	参数列表		

➤ **响应**

返回合约的数据结果

2.10.2 说明

返回参数

2.11.查看区块列表

接口地址：GET 方法 /:alliance/:channel/blocks

例子：http://apibass.hnsoft.com/alliance/channel/blocks

2.11.1 参数

➤ **URL 请求**

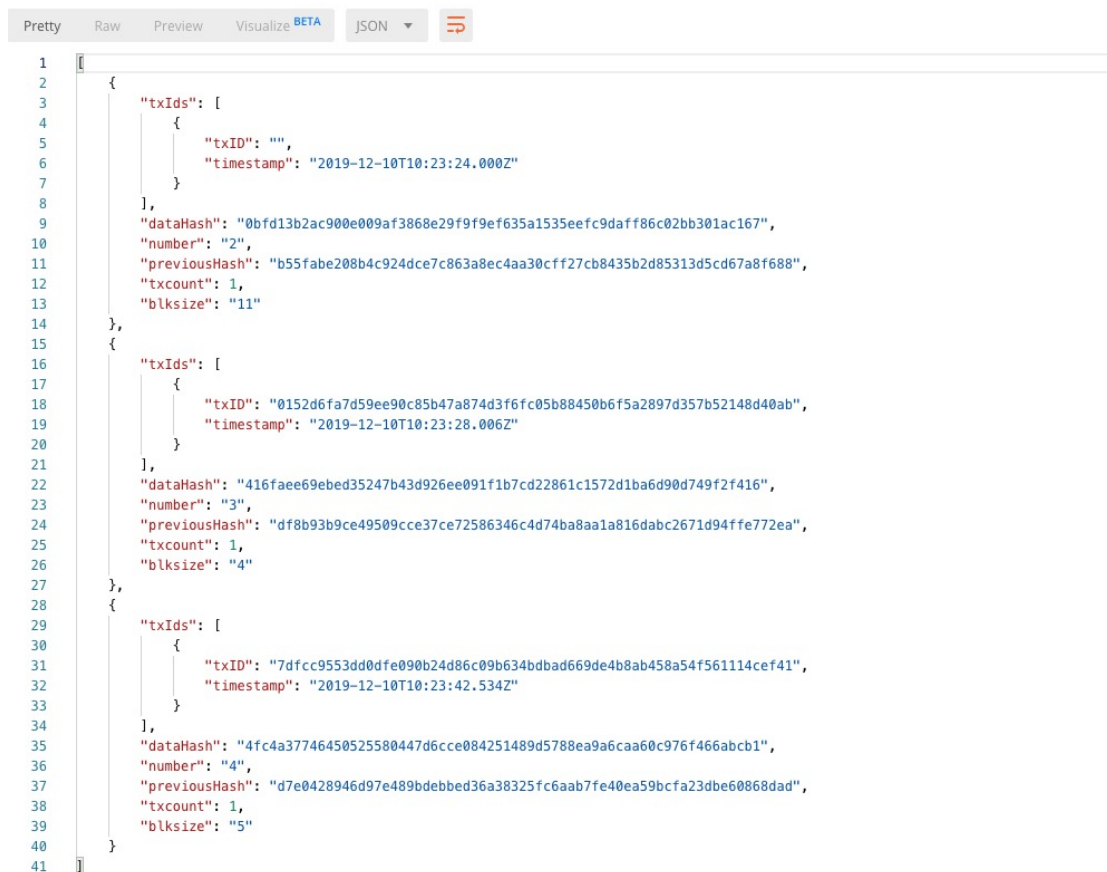
参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		

➤ **body 请求**

参数	参数名称	参数说明	样例
----	------	------	----

from	起始值		
to	结束值		

➤ 响应



2.11.2 说明

返回参数

blksize 单位为 KB

2.12.查看交易列表

接口地址: GET 方法 /:alliance/:channel/:org/queryByDataRange

例子: <http://apibass.hnsoft.com/alliance/channel/org/queryByDataRange>

2.12.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ 请求

参数	参数名称	参数说明	样例
startKey	起始值		
endKey	结束值		

➤ 响应

```
Pretty Raw Preview Visualize BETA JSON
1  {
2    {
3      "Key": "1",
4      "Record": {
5        "ID": "1",
6        "data": "008999222111122343434",
7        "owner": "daiqunbiao",
8        "txid": "cb0e5ea1abb19f0d27782d42b7fb037ea5b16774c6026e33e3dc33d6704efe74"
9      }
10   },
11   {
12     "Key": "2",
13     "Record": {
14       "ID": "2",
15       "data": "008999222111122343434",
16       "owner": "daiqunbiao",
17       "txid": "c53e34144263ed683ce625d8a18e8d3248b1d68223d1a8ab42d440453e13c511"
18     }
19   },
20   {
21     "Key": "3",
22     "Record": {
23       "ID": "3",
24       "data": "008999222111122343434",
25       "owner": "daiqunbiao",
26       "txid": "4be59684f4541898e296a611b3d4422577c7b3ccea493f7e466a4b0f3c909c27"
27     }
28   },
29   {
30     "Key": "4",
31     "Record": {
32       "ID": "4",
33       "data": "008999222111122343434",
34       "owner": "daiqunbiao",
35       "txid": "4a20143cb2ea7bbada432ba44b78c2ee23cb65d9d30c45e93ecbde50945b4b2"
36     }
37   }
}
```

2.12.2 说明

返回参数

2.13.删除账户

接口地址: Get 方法 /:alliance/:channel/users/revokeUser

例子: <http://apibass.hnsoft.com/alliance/channel/users/revokeuser>

2.13.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		

➤ **body 请求**

参数	参数名称	参数说明	样例
username	用户名		

➤ **响应**

成功返回信息的示例如下：

失败返回信息的示例如下：

2.13.2 说明

返回参数

2.14.查看通道区块高度

接口地址：GET 方法 /:alliance/:channel/height

例子：http://apibass.hnsoft.com/alliance/channel/height

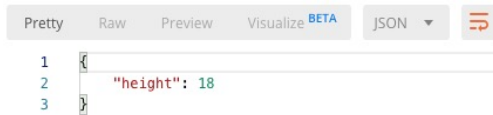
2.14.1 参数

➤ **URL 请求**

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		

➤ **响应**

成功返回信息的示例如下：



2.14.2 说明

返回参数

2.15.查看通道的 Peer 节点数

接口地址：GET 方法 /:alliance/:channel/peers

例子：http://apibass.hnsoft.com/:alliance/:channel/peers

2.15.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		

➤ 响应

成功返回信息的示例如下：

```
Pretty Raw Preview Visualize BETA JSON
1  [
2    {
3      "mspID": "Org1MSP",
4      "name": "peer0.org1.example.com:7051",
5      "url": "grpcs://peer0.org1.example.com:7051"
6    },
7    {
8      "mspID": "Org1MSP",
9      "name": "peer1.org1.example.com:8051",
10     "url": "grpcs://peer1.org1.example.com:8051"
11   },
12   {
13     "mspID": "Org2MSP",
14     "name": "peer0.org2.example.com:9051",
15     "url": "grpcs://peer0.org2.example.com:9051"
16   },
17   {
18     "mspID": "Org2MSP",
19     "name": "peer1.org2.example.com:10051",
20     "url": "grpcs://peer1.org2.example.com:10051"
21   }
22 ]
```

2.15.2 说明

返回参数

2.16.查看通道的 order 节点数

接口地址：GET 方法 `/:alliance/:channel/orderers`

例子：`http://apibass.hnsoft.com/:alliance/:channel/orderers`

2.16.1 参数

➤ 请求

参数	参数名称	参数说明	样例
alliance	网络名称		
channel	通道名称		

➤ 响应

成功返回信息的示例如下：

```
Pretty Raw Preview Visualize BETA JSON
1 [
2   {
3     "name": "orderer1.example.com:7050",
4     "url": "grpc://orderer1.example.com:7050"
5   }
6 ]
```

2.16.2 说明

返回参数

2.17.查看通道的链码数

接口地址：GET 方法 `/:alliance/:channel/chaincodes`

例子：`http://apibass.hnsoft.com/:alliance/:channel/chaincodes`

2.17.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		

➤ 响应

成功返回信息的示例如下：

```
Pretty Raw Preview Visualize BETA JSON ↕
1  {
2    "chaincodes": [
3      {
4        "name": "huobicc",
5        "version": "1.0",
6        "path": "github.com/chaincode/baas/go/",
7        "input": "",
8        "escc": "escc",
9        "vscc": "vscc",
10       "id": {
11         "buffer": {
12           "type": "Buffer",
13           "data": []
14         },
15         "offset": 0,
16         "markedOffset": -1,
17         "limit": 0,
18         "littleEndian": false,
19         "noAssert": false
20       }
21     },
22     {
23       "name": "mycc",
24       "version": "1.0",
25       "path": "github.com/chaincode/sample/go/",
26       "input": "",
27       "escc": "escc",
28       "vscc": "vscc",
29       "id": {
30         "buffer": {
31           "type": "Buffer",
32           "data": []
33         },
34         "offset": 0,
35         "markedOffset": -1,
36         "limit": 0,
37         "littleEndian": false,
38         "noAssert": false
39       }
40     }
41   ]
42 }
```

2.17.2 说明

返回参数

2.18.查看通道信息

接口地址: GET 方法 /:alliance/:channel/info

例子: <http://apibass.hnsoft.com/:alliance/:channel/info>

2.18.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		

➤ 响应

成功返回信息的示例如下：

```
Pretty Raw Preview Visualize BETA JSON ↕
1  {
2    "height": {
3      "low": 18,
4      "high": 0,
5      "unsigned": true
6    },
7    "currentBlockHash": {
8      > "buffer": { ...
82    },
83    "offset": 4,
84    "markedOffset": -1,
85    "limit": 36,
86    "littleEndian": true,
87    "noAssert": false
88  },
89  "previousBlockHash": {
90    > "buffer": { ...
164  },
165  "offset": 38,
166  "markedOffset": -1,
167  "limit": 70,
168  "littleEndian": true,
169  "noAssert": false
170  }
171 }
```

2.18.2 说明

返回参数

2.19.查看通道的 Peer 节点数

接口地址：GET 方法 /:alliance/:channel/peers

例子：http://apibass.hnsoft.com/:alliance/:channel/peers

2.19.1 参数

➤ url 参数

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		

➤ 响应

成功返回信息的示例如下：

```
[
  {
    "mspID": "Org2MSP",
    "name": "peer1.org2.example.com:10051",
    "url": "grpcs://localhost:10051"
  },
  {
    "mspID": "Org2MSP",
    "name": "peer0.org2.example.com:9051",
    "url": "grpcs://localhost:9051"
  },
  {
    "mspID": "Org1MSP",
    "name": "peer1.org1.example.com:8051",
    "url": "grpcs://localhost:8051"
  },
  {
    "mspID": "Org1MSP",
    "name": "peer0.org1.example.com:7051",
    "url": "grpcs://localhost:7051"
  }
]
```

2.19.2 说明

返回参数

2.20.查看区块信息通过 Hash

接口地址：GET 方法 /:alliance/:channel/block/hash

例子: <http://apibass.hnsoft.com/:alliance/:channel/block/hash/0fe56f44727295c1bb28b627f3b00915f0f121b37f5daccaad6c8ae917db667b>

2.20.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: blockhash	区块Hash		

➤ 响应

成功返回信息的示例如下:

```
Pretty  Raw  Preview  Visualize BETA  JSON  展开
```

```
1  {
2    "header": {
3      "number": "1",
4      "previous_hash": "13e07b579c14bd5e27d1526017d46aa3adc4c93e09697f95a14fdfcc4d844e83",
5      "data_hash": "6079b42384250eed56f497663a6fd000af08122c71505b85f1255da16d6c3e34"
6    },
7    "data": { ...
1102  },
1103    "metadata": {
1104      "metadata": [
1105        {
1106          "value": "\n\b",
1107          "signatures": [ ...
1221        ],
1222      ],
1223      {
1224        "value": {
1225          "index": "1"
1226        },
1227        "signatures": []
1228      },
1229      [
1230        0
1231      ]
1232    ]
1233  }
1234 }
```

2.20.2 说明

返回参数

2.21.查看区块头信息

接口地址：GET 方法 /:alliance/:channel/header/:blockNumber

例子：http://apibass.hnsoft.com/alliance/channel/header/1

2.21.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: blockNumber	区块号		

➤ 响应

成功返回信息的示例如下：

Pretty	Raw	Preview	Visualize BETA	JSON	
1					
2					"number": "2",
3					"previousHash": "b90f9fc1b46a5ac58d15f968b7a8d72469c550b9fc50620d5fb50f722d911cd9",
4					"dataHash": "559b9a493d2630f3c8dd53b729d8d9db02c8581931447e23982cdbc90f0bf395"
5					

2.21.2 说明

返回参数

2.22.通过区块号获取区块信息

接口地址：GET 方法 /:alliance/:channel/block/:blockNumber

例子：http://apibass.hnsoft.com/alliance/channel/block/2

2.22.1 参数

➤ 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: block	区块号		

➤ 响应

成功返回信息的示例如下：

```
Pretty  Raw  Preview  Visualize BETA  JSON  ↺
1  {
2    "header": {
3      "number": "2",
4      "previous_hash": "b90f9fc1b46a5ac58d15f968b7a8d72469c550b9fc50620d5fb50f722d911cd9",
5      "data_hash": "559b9a493d2630f3c8dd53b729d8d9db02c8581931447e23982cdbc90f0bf395"
6    },
7    "data": {
8      "data": [
9        {
10 >       "signature": {--
84 >       },
85 >       "payload": {--
1110 >       }
1111     ]
1112   },
1113   "metadata": {
1114     "metadata": [
1115       {
1116         "value": "\n\b",
1117         "signatures": [--
1232 >       ]
1233     },
1234     {
1235       "value": {
1236         "index": "2"
1237       },
1238       "signatures": []
1239     },
1240     [
1241       0
1242     ]
1243   ]
1244 }
1245 }
```

2.22.2 说明

返回参数

2.23.安装链码

接口地址：POST 方法 `/:alliance/:channel/chaincode/install`

例子：`http://apibass.hnsoft.com/devbernet01/mychannel/chaincode/install`

2.23.1 参数

➤ url 参数

参数	参数名称	参数说明	样例
<code>:alliance</code>	联盟名称		dev-cybernet01
<code>:channel</code>	通道名称	目标通道名称	mychannel

➤ form-data 参数

参数	参数名称	参数说明	样例
chaincode	合约代码文件	参数类型文件	
chaincodeID	合约唯一名称		Sample11
orgName	组织名称		Org1
chaincodeVersion	合约版本		1
chaincodeType	合约语言类型	支持go/java/node	node
targets	安装目标节点列表		peer1.org1.example.com

➤ 响应

成功返回信息的示例如下：

```
{
  "success": true,
  "message": "Successfully installed chaincode"
}
```

2.24.通过交易 ID 获取历史数据

接口地址：GET 方法 `/:alliance/:channel/:org/getHistoryForData/:id`

例子：

`http://apibass.hnsoft.com/alliance/channel/org1/getHistoryForData/2`

2.24.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ body 请求

参数	参数名称	参数说明	样例
id	交易ID		

➤ 响应

成功返回信息的示例如下：

Pretty	Raw	Preview	Visualize BETA	JSON	🔗
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					

```
{
  "TxId": "87a21b93bd7d7a28e94e001dd24185899cac6fbe78d49237d8d592a3d28ea651",
  "Value": {
    "owner": "daiqunbiao",
    "ID": "7",
    "txid": "87a21b93bd7d7a28e94e001dd24185899cac6fbe78d49237d8d592a3d28ea651",
    "data": "00899922211122343434"
  },
  "Timestamp": "2019-12-21 10:53:28.659 +0000 UTC",
  "IsDelete": "false"
}
```

2.24.2 说明

返回参数

2.25.通过 owner 获取交易数据

接口地址：GET 方法 /:alliance/:channel/:org/queryDataByOwner/:owner

例子：

http://apibass.hnsoft.com/alliance/channel/org1/queryDataByOwner/daiqunbiao

2.25.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ body 请求

参数	参数名称	参数说明	样例
owner	拥有者		

➤ 响应

成功返回信息的示例如下：

```
1  [
2    {
3      "Key": "1",
4      "Record": {
5        "ID": "1",
6        "data": "008999222111122343434",
7        "owner": "daiqunbiao",
8        "txid": "cb0e5ea1abb19f0d27782d42b7fb037ea5b16774c6026e33e3dc33d6704efe74"
9      }
10   },
11   {
12     "Key": "2",
13     "Record": {
14       "ID": "2",
15       "data": "008999222111122343434",
16       "owner": "daiqunbiao",
17       "txid": "c53e34144263ed683ce625d8a18e8d3248b1d68223d1a8ab42d440453e13c511"
18     }
19   },
20   {
21     "Key": "3",
22     "Record": {
23       "ID": "3",
24       "data": "008999222111122343434",
25       "owner": "daiqunbiao",
26       "txid": "4be59684f4541898e296a611b3d4422577c7b3ccea493f7e466a4b0f3c909c27"
27     }
28   },
29   { ...
37   },
38   { ...
46   },
47   { ...
55   },
56   {
57     "Key": "7",
58     "Record": {
59       "ID": "7",
60       "data": "008999222111122343434",
61       "owner": "daiqunbiao",
62       "txid": "87a21b93bd7d7a28e94e001dd24185899cac6fbe78d49237d8d592a3d28ea651"
63     }
64   },
65   {
66     "Key": "8",
67     "Record": {
68       "ID": "8",
69       "data": "008999222111122343434",
70       "owner": "daiqunbiao",
71       "txid": "e10c8542ce3e2137eec6ec3e85a7beedf44b28ed588130e2110ab7f6dbfa2087"
72     }
73   }
74 ]
```

2.25.2 说明

返回参数

2.26. 获取最近的交易列表

接口地址: GET 方法 /:alliance/:channel/getLatestTxList

/:maxTx

例子: <http://apibass.hnsoft.com/alliance/channel/getLatestTxList/40>

2.26.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: maxTx	最大的交易数		

➤ body 请求

参数	参数名称	参数说明	样例

➤ 响应

成功返回信息的示例如下:

```
Pretty Raw Preview Visualize BETA JSON
1
2 {
3   "txIds": [
4     {
5       "txID": "61512e844ba93a5da4bfc01aa94970c75ada3b283361ea6adeaa163479b8862f",
6       "timestamp": "2019-12-30T09:07:49.507Z"
7     }
8   ],
9   "dataHash": "87f6074b0c4e2148b21a42f52be815934b0f7def68cd2428b49dd9258d470f1f",
10  "number": "14",
11  "previousHash": "86343f883fb81d1a78bed87e263681f64f13fa8290b0099f3c7ce7ac1dc77744",
12  "txcount": 1,
13  "blksize": "5"
14 },
15 {
16   "txIds": [
17     {
18       "txID": "be8e19c5f0bebd0d648e8057e9de3d71384e74461077f5b6dbdde321b4a7b0d",
19       "timestamp": "2019-12-30T09:01:26.003Z"
20     }
21   ],
22   "dataHash": "9cbc9d90266c5a586a19ec9707e184c677d5e810a4243140f125a5a9e91d231e",
23   "number": "13",
24   "previousHash": "32fdacd10b430c1c201fa9623c36a66b864e9d4988b3f96f3827b571159a7d15",
25   "txcount": 1,
26   "blksize": "5"
27 }
28 }
```

2.26.2 说明

返回参数

2.27.通过交易 ID 获取交易信息

接口地址：GET 方法 `/:alliance/:channel/getTransactionByID`
`/:txnID`

例子：`http://apibass.hnsoft.com/alliance/channel/getTransactionByID`
`/2cd21d8359158b7653ed685ffa62fa5a90673628db4b392c627d06ad8378f6fa`

2.27.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: txnID	交易ID		

➤ body 请求

参数	参数名称	参数说明	样例

➤ 响应

成功返回信息的示例如下：

```
1 {
2   "validationCode": 0,
3   "transactionEnvelope": {
4     "signature": {
5       "type": "Buffer",
6       "data": [~
7     ],
8   },
9   "payload": {
10    "header": {
11      "channel_header": {
12        "type": 3,
13        "version": 1,
14        "timestamp": "2019-12-30T10:17:58.884Z",
15        "channel_id": "mychannel",
16        "tx_id": "2cd21d8359158b7653ed685ffa62fa5a90673628db4b392c627d06ad8378f6fa",
17        "epoch": "0",
18        "extension": {
19          "type": "Buffer",
20          "data": [~
21        ],
22      },
23      "typeString": "ENDORSE_TRANSACTION"
24    },
25    "signature_header": {~
26  },
27  "data": {
28    "actions": [
29      {
30        "header": {~
31      },
32      "payload": {~
33    ]
34  }
35 }
```

2.27.2 说明

返回参数

2.28.获取网络的所有节点

接口地址：GET 方法 /:alliance/peers

例子：http://apibass.hnsoft.com/alliance/peers

2.28.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		

➤ 响应

成功返回信息的示例如下：

```
Pretty  Raw  Preview  Visualize BETA  JSON    
1  {  
2    {  
3      "mspID": "Org1MSP",  
4      "name": "peer0.org1.example.com:7051",  
5      "url": "grpcs://peer0.org1.example.com:7051"  
6    },  
7    {  
8      "mspID": "Org2MSP",  
9      "name": "peer0.org2.example.com:9051",  
10     "url": "grpcs://peer0.org2.example.com:9051"  
11   }  
12 }
```

2.28.2 说明

返回参数

2.29. 获取网络所有的智能合约

接口地址：GET 方法 /:alliance/chaincodes

例子：http://apibass.hnsoft.com/alliance/chaincodes

2.29.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		

➤ 响应

成功返回信息的示例如下：

```
Pretty Raw Preview Visualize BETA JSON   
1 {  
2   "name": huobicc, version: 1.0, path: github.com/chaincode/baas/go/",  
3   "name": tokencc, version: 1.0, path: github.com/chaincode/jihong/go/"  
4 }
```

2.29.2 说明

返回参数

3 积分商城 Demo 智能合约 V1 接口

3.1.创建账户

接口地址：POST 方法

`/:alliance/:channel/:org/chaincode/token/createAccount`

例子：

`http://apibass.hnsoft.com/alliance/channel/org/chaincode/token/createAccount`

3.1.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	网络名称		

➤ body 请求

参数	参数名称	参数说明	样例
accountName	账户名称	商家账户	伊利、蒙牛

➤ 响应

成功返回信息的示例如下：

```
Pretty Raw Preview Visualize BETA JSON 
```

```
1 {
2   "Name": "daiqunbiao3",
3   "Frozen": false,
4   "BalanceOf": {}
5 }
```

3.1.2 说明

返回参数

3.2.积分发行

接口地址：POST 方法

`/:alliance/:channel/:org/chaincode/token/initToken`

例子：

`http://apibass.hnsoft.com/alliance/channel/0rg1/chaincode/token/initToken`

3.2.1 参数

➤ URL 请求

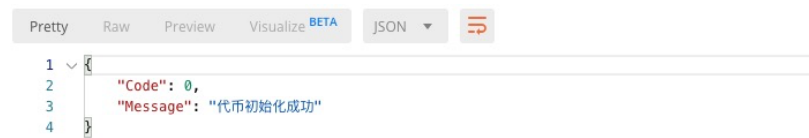
参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	网络名称		

➤ body 请求

参数	参数名称	参数说明	样例
accountName	账户名称	发行该币的商家账户	吉宏、伊利、蒙牛
tokenName	积分名称		JiHong
tokenSymbol	积分简称		JH
tokenSupply	积分发行量		100000000
tokenValue	积分价值		1
tokenUsage	积分用途		

➤ 响应

成功返回信息的示例如下：



3.2.2 说明

返回参数

3.3.积分增发

接口地址：POST 方法

`/:alliance/:channel/:org/chaincode/token/mintToken`

例子：

`http://apibass.hnsoft.com/alliance/channel/0rg1/chaincode/token/mintToken`

3.3.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ body 请求

参数	参数名称	参数说明	样例
----	------	------	----

accountName	账户名称	发行该币的商家账户	吉宏、伊利、蒙牛
tokenName	积分名称		JiHong
amount	增发量		10000

➤ 响应

成功返回信息的示例如下：



3.3.2 说明

返回参数

3.4.积分销毁

接口地址：POST 方法

`/:alliance/:channel/:org/chaincode/token/burnToken`

例子：

`http://apibass.hnsoft.com/alliance/channel/0rg1/chaincode/token/burnToken`

3.4.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ body 请求

参数	参数名称	参数说明	样例
accountName	账户名称	发行该币的商家账户	吉宏、伊利、蒙牛
tokenName	积分名称		JiHong
amount	销毁量		10000

➤ 响应

成功返回信息的示例如下：



3.4.2 说明

返回参数

3.5.积分通兑

接口地址：POST 方法

`/:alliance/:channel/:org/chaincode/token/exchangeToken`

例子：

`http://apibass.hnsoft.com/alliance/channel/Org1/chaincode/token/exchangeToken`

3.5.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		

: channel	通道名称		
: org	组织名称		

➤ body 请求

参数	参数名称	参数说明	样例
fromAccount	账户名称	发起通兑的商家账户	伊利
fromToken	积分名称	兑换币	YiLi
toAccount	账户名称	被通兑的商家账户	吉宏
toToken	积分名称	被兑换币	JiHong
amount	兑换量	From量	10000
rate	兑换率	1个from兑rate个to	10

➤ 响应

成功返回信息的示例如下：

Pretty	Raw	Preview	Visualize BETA	JSON	
1	{				
2	"Code": 0,				
3	"Message": "兑换成功"				
4	}				

3.5.2 说明

返回参数

3.6.账户单币种余额查询

接口地址：GET 方法 /:alliance/:channel/:org/chaincode/token/balance

例子:

`http://apibass.hnsoft.com/alliance/channel/0rg1/chaincode/token/
balance`

3.6.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ body 请求

参数	参数名称	参数说明	样例
accountName	账户名称	要查询的账户	伊利
tokenName	积分名称	账户下的某币种	YiLi or JiHong

➤ 响应

成功返回信息的示例如下:

Pretty	Raw	Preview	Visualize BETA	JSON ▼	
1	{				
2	"IPX": {				
3	"Lock": false,				
4	"Balance": 7093219574				
5	}				
6	}				

3.6.2 说明

返回参数

3.7.账户余额查询

接口地址：GET 方法

`/:alliance/:channel/:org/chaincode/token/balanceAll`

例子：

`http://apibass.hnsoft.com/alliance/channel/Org1/chaincode/token/balanceAll`

3.7.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	网络名称		

➤ body 请求

参数	参数名称	参数说明	样例
accountName	账户名称	要查询的账户	伊利

➤ 响应

成功返回信息的示例如下：

```
Pretty  Raw  Preview  Visualize BETA  JSON  ▼  
```

```
1  {
2    "IPX": {
3      "Lock": false,
4      "Balance": 7093219574
5    },
6    "IPX2": {
7      "Lock": false,
8      "Balance": 9000000000
9    },
10   "IPX5": {
11     "Lock": false,
12     "Balance": 6993220000
13   },
14   "eth": {
15     "Lock": false,
16     "Balance": 5000
17   }
18 }
```

3.7.2 说明

返回参数

3.8.积分详情查询

接口地址：GET 方法 /:alliance/:channel/:org/chaincode/token/showToken

例子：

<http://apibass.hnsoft.com/alliance/channel/0rg1/chaincode/token/showToken>

3.8.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
----	------	------	----

: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ **body 请求**

参数	参数名称	参数说明	样例
tokenName	积分名称	要查询的积分类型	JiHong

➤ **响应**

成功返回信息的示例如下：

3.8.2 说明

返回参数

3.9.交易记录查询

接口地址：GET 方法

/:alliance/:channel/:org/chaincode/token/showAccountHistory

例子：

http://apibass.hnsoft.com/alliance/channel/Org1/chaincode/token/showAccountHistory

3.9.1 参数

➤ **URL 请求**

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ body 请求

参数	参数名称	参数说明	样例
accountName	账户名称	要查询的账户名	吉宏
tokenName	积分名称	要查询的积分类型	BitCoin
txType	交易类型	增发、销毁、通兑	mint/burn/exchange

➤ 响应

成功返回信息的示例如下：



3.9.2 说明

返回参数

3.10.所有兑换交易记录查询

接口地址：GET 方法

/:alliance/:channel/:org/chaincode/token/showAllExchangeHistory

例子：

http://apibass.hnsoft.com/alliance/channel/Org1/chaincode/token/showAllExchangeHistory

3.10.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
----	------	------	----

: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ body 请求

参数	参数名称	参数说明	样例
----	------	------	----

➤ 响应

成功返回信息的示例如下：

```
Pretty  Raw  Preview  Visualize BETA  JSON    
1  {  
2    "daiqunbiao1-btc": [  
3      {  
4        "Id": 0,  
5        "TxId": "ab6d0ea8be0363a27ecbb134aac0b1553254fb3489a61a30ff651fd2e55bb3b1",  
6        "From": "daiqunbiao1",  
7        "FromToken": "btc",  
8        "To": "daiqunbiao2",  
9        "ToToken": "ht",  
10       "AmountFrom": 500,  
11       "AmountTo": 5000,  
12       "timestamp": 1579146640  
13     }  
14   ]  
15 }
```

3.10.2 说明

返回参数

3.11.某账户发行的所有积分情况

接口地址：Post 方法

`/:alliance/:channel/:org/chaincode/token/showTokensOfAccount`

例子：

`http://apibass.hnsoft.com/alliance/channel/0rg1/chaincode/token/showTokensOfAccount`

3.11.1 参数

➤ URL 请求

参数	参数名称	参数说明	样例
: alliance	网络名称		
: channel	通道名称		
: org	组织名称		

➤ body 请求

参数	参数名称	参数说明	样例
accountName	账户名称	要查询的积分发行账户	JiHong

➤ 响应

成功返回信息的示例如下：

Pretty	Raw	Preview	Visualize BETA	JSON	
1	{				
2	{				
3	"Lock": false,				
4	"Owner": "daiqunbiao2",				
5	"TokenName": "ht",				
6	"TokenSymbol": "eth002",				
7	"TokenValue": 2000,				
8	"TotalSupply": 20000000,				
9	"InitSupply": 20000000,				
10	"TokenUsage": "",				
11	"IssueTime": 1579146623				
12	}				
13	}				

3.11.2 说明

返回参数