# Asset transfer Scenario using multiple addresses on your node

## Preparation

# You have already created your Multichain Instance (refer to “Create your first Multichain Node”) or you have even created your own network of multichain nodes (refer to “Build up your own Multichain network”). You have noted down your api key(s).

## Execution

Your Multichain node supports multiple addresses. The addresses themselves will not be replicated across the network since they are bound to your node’s wallet, that is managed by your Multichain instance (e.g. storing the private key). You will therefore make sure to send transaction for your specific address to the right node, the one who contains the wallet for your address.

You may use multiple addresses for a scenario where you have n validating node, but m participants that want to access data only, and m>>n. For example, a group of banks as validating nodes and their customers that want to send transaction across. Each customer can be assigned to exactly one bank and have one address on a node of the bank.

We will call json-rpc to the node. For this you might use any http client software or library of your programming language of choice. For demos, we use Postman or Postman for Chrome. The reference guide for json-rpc commands in Multichain is found under:

<https://www.multichain.com/developers/json-rpc-api/>

### Create an asset

As a first step, let’s create an asset. We name our asset SAPCoins.

Execute the following json-rpc call to your node (host and api key is given in the service key of the Multichain service):

POST HTTP/1.1

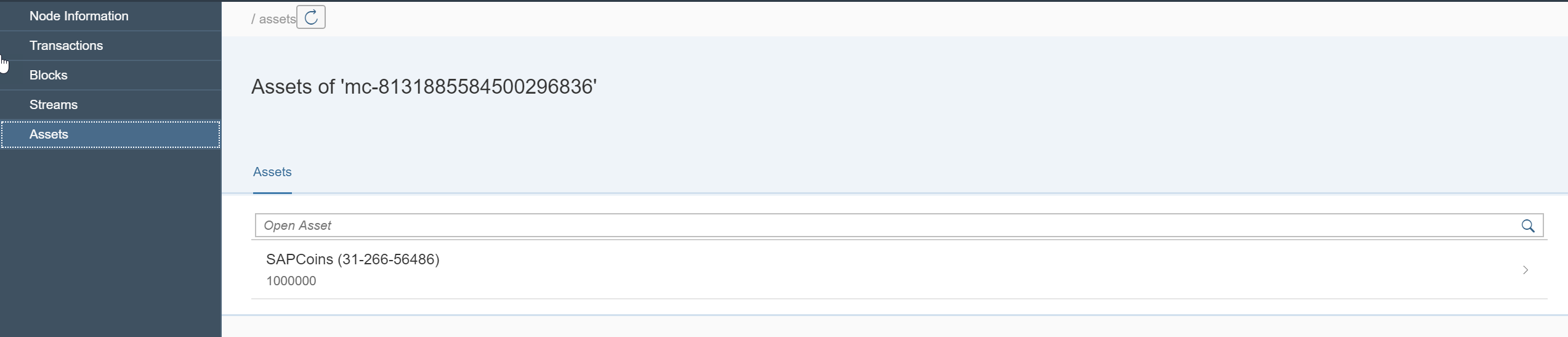
Host: maas-proxy.cfapps.sap.hana.ondemand.com/<instance ID>/rpc

(HEADER) apikey: <your API Key>

(BODY) {"method": "issue", "params": ["<address of the issuer>", "SAPCoins", 1000000] }

The address of the issuer can be found in the Multichain dashboard of your node.

If the http request was successful, you will find the asset on your dashboard:



### Create a new address

We want to transfer some of the SAP Coins now to another address, e.g. a customer of the bank. We will create the new address on the same node, but it will work the same if you create the new address on another node (e.g. simulating a new bank receiving a cross-bank payment).

Execute the following json-rpc call to your node:

POST HTTP/1.1

Host: maas-proxy.cfapps.sap.hana.ondemand.com/<instance ID>/rpc

(HEADER) apikey: <your API Key>

(BODY) {"method": "getnewaddress", "params": []}

In the response, you get a new address. This address with its private key is stored on the node’s wallet. If you prefer to manage private keys on your own you may go for non-wallet addresses in Multichain.

You can list the addresses using the rpc method getaddresses. We say the primary address is the one generated while creating the node (in this example 1A1YXCibRKsTFDGBrfJPNtMTbBVfMBzwJSbmsK), and we say the just generated address is the second one (in this example 1HHVYGc4GtmL2SvMvCtV24T6gYBvqq3MfcXNer).

The second address does not yet have any permissions, therefore we need to grant it. Our first address will grant permissions to the second one. Execute the following command:

POST HTTP/1.1

Host: maas-proxy.cfapps.sap.hana.ondemand.com/<instance ID>/rpc

(HEADER) apikey: <your API Key>

(BODY) {"method": "grant", "params": ["<address >", "connect,send,receive,issue "] }

As address you enter the second address (1HHVYGc4GtmL2SvMvCtV24T6gYBvqq3MfcXNer). The method grant will use the primary address as the grantee. If you want to explicitly use an address for granting permissions, take the command grantfrom.

You second will now have permissions to receive and send assets.

### Send assets

Our primary address has 1000000 SAP Coins while the second node has no coins yet. Let’s be generous and send 5 SAPCoins from the primary address to the second one.

Execute the following json-rpc call to your node:

POST HTTP/1.1

Host: maas-proxy.cfapps.sap.hana.ondemand.com/<instance ID>/rpc

(HEADER) apikey: <your API Key>

(BODY) {"method": "sendasset", "params": ["<address>", "SAPCoins", 5]]}

As address enter the second address. The method sendasset will use the primary address as sender address. If you want to explicitly use a sender address, take the command sendassetfrom.

The 5 SAPCoins are successfully transferred to the second address. You may check the balances of your addresses using the command getaddressbalances.

Execute the following json-rpc call to your node:

POST HTTP/1.1

Host: maas-proxy.cfapps.sap.hana.ondemand.com/<instance ID>/rpc

(HEADER) apikey: <your API Key>

(BODY) {"method": "getaddressbalances", "params": ["<address>"]}

For the second address you get the response:

{ "result": [ {

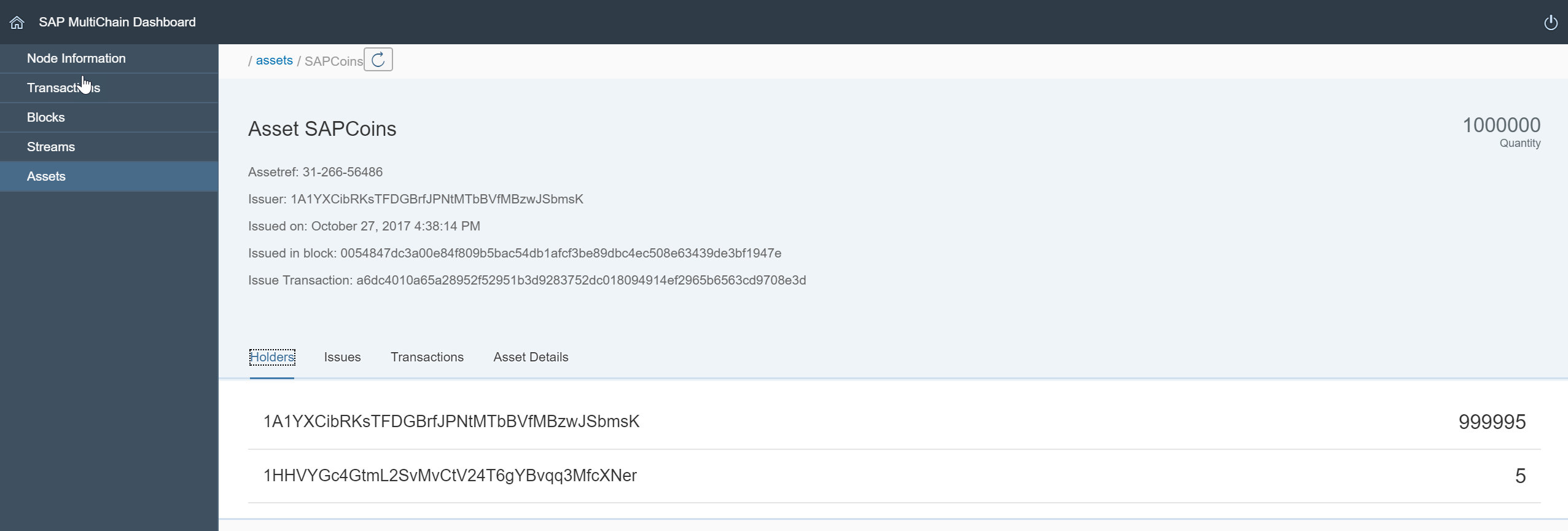
"name": "SAPCoins",

"assetref": "31-266-56486",

"qty": 5

}] }

Alternatively, you will also find asset information in the SAP Multichain Dashboard in your SAP Cloud Platform Multichain service:



You have successfully implemented your first asset transfer scenario. As an extension you may create an asset on a second node and transfer SAPCoins to it as well.