How we can send a message to the IOTA Network?

First, we must setup the environment using the command line (for Windows OS) or terminal (for MacOS or Linux OS):

* We create a folder (for example, iota-app) for our application

mkdir iota-app

cd iota-app

* We initialize the Node.js project:

npm init

* We install the IOTA core client libraries:

npm install @iota/core @iota/converter

* We run the script.js file (below) from node and we will obtain a hash string (like KJQLEPCOIOHIQEZMO9MLHZEIAWHGZZSFJBRREKMI9XZGJJDIQFVZPRCRZTKJHFVCBEIOIULARRRMHO999):

node script.js

where the file script.js is below:

// script.js

const Iota = require('@iota/core');

const Converter = require('@iota/converter');

// Connect to a node

const iota = Iota.composeAPI({

 provider: 'https://nodes.devnet.iota.org:443'

});

const depth = 3;

const minimumWeightMagnitude = 9;

// Define a seed and an address.

// These do not need to belong to anyone or have IOTA tokens.

// They must only contain a mamximum of 81 trytes

// or 90 trytes with a valid checksum

const address = 'HEQLOWORLDHELLOWORLDHELLOWORLDHELLOWORLDHELLOWORLDHELLOWORLDHELLOWORLDHELLOWOR99D';

const seed = 'PUEOTSEITFEVEWCWBTSIZM9NKRGJEIMXTULBACGFRQK9IMGICLBKW9TTEVSDQMGWKBXPVCBMMCXWMNPDX';

// Define a message to send.

// This message must include only ASCII characters.

const message = JSON.stringify({"message": "Hello, world! Today is June 15, 2021, and we are at Blocks Summer School!"});

// Convert the message to trytes

const messageInTrytes = Converter.asciiToTrytes(message);

// Define a zero-value transaction object

// that sends the message to the address

const transfers = [

 {

 value: 0,

 address: address,

 message: messageInTrytes

 }

];

// Create a bundle from the `transfers` array

// and send the transaction to the node

iota

 .prepareTransfers(seed, transfers)

 .then(trytes => {

 return iota.sendTrytes(trytes, depth, minimumWeightMagnitude);

 })

 .then(bundle => {

 console.log(bundle[0].hash);

 })

 .catch(err => {

 console.error(err)

 });

* We can find, using the hash string, the message that we have sent in the <https://explorer.iota.org/legacy-devnet>:

<https://explorer.iota.org/legacy-devnet/transaction/KJQLEPCOIOHIQEZMO9MLHZEIAWHGZZSFJBRREKMI9XZGJJDIQFVZPRCRZTKJHFVCBEIOIULARRRMHO999>

In the last version of IOTA, sending data to the tangle is more easier.

const { ClientBuilder } = require('@iota/client');

const client = new ClientBuilder().network('testnet').build();

const response = client.message().index('Example').data('Hello from IOTA!').submit();

The variable *response* will contain the following structure format:

Promise {

 {

 message: {

 networkId: '14379272398717627559',

 parentMessageIds: [Array],

 payload: [Object],

 nonce: '4611686018427618467'

 },

 messageId: 'ad97382c289a53490021188c600c807490432b90ce6728ba8e4ad3395a1491cd'

 }

}