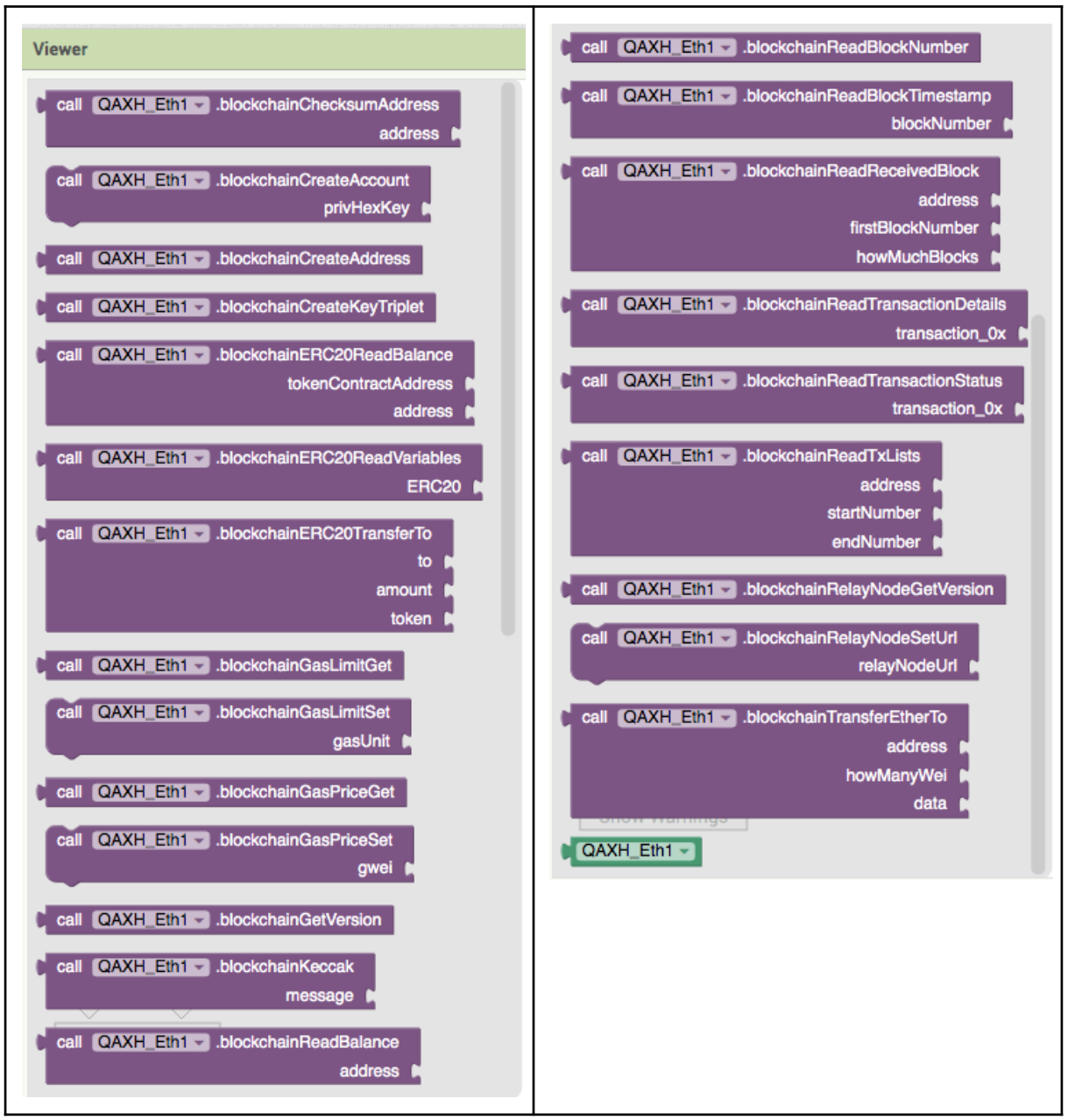
Qaxh_ETH 108 documentation

List of AI2 blocks



.blockchainChecksumAddress

Return the address, but "checksummed", as defined in the Ethereum standard.

Errors : those returned by the org.web3j.crypto.Keys.toChecksumAddress function.

.blockchainCreateAccount

Instantiate the extension private key and nonce value internal variables. This function must be called before using the other functions of the extension.

 This function must be called only once.

.blockchainCreateAddress

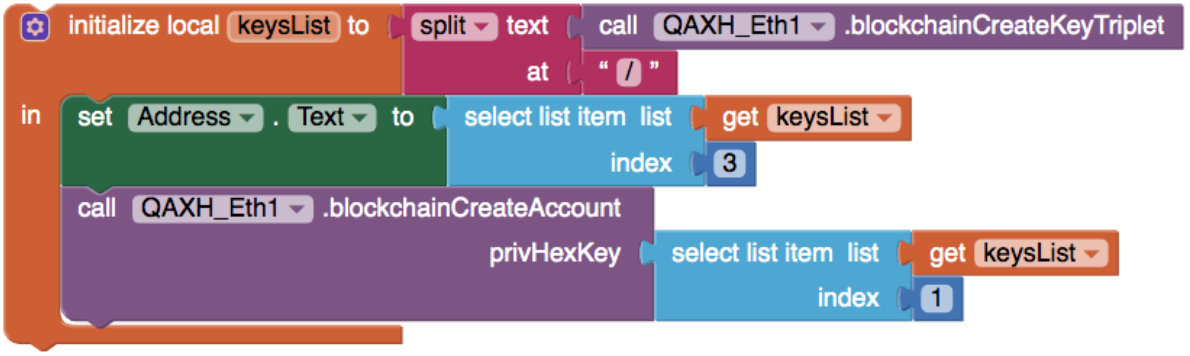
Generates an ethereum private/public key pair. The results is in the following format:

“PrivateKey/AddressKey”.

.blockchainCreateKeyTriplet

Generates an ethereum private/public key/address. The results is in the following format:
“PrivateKey/PublicKey/AddressKey”.

Here is an example of creation and management of keys.



The function returns the three keys separated by “/”.

Note that you should apply the .toChecksumAddress function on the AddressKey in order for it to be valid.

.blockchainERC20ReadBalance

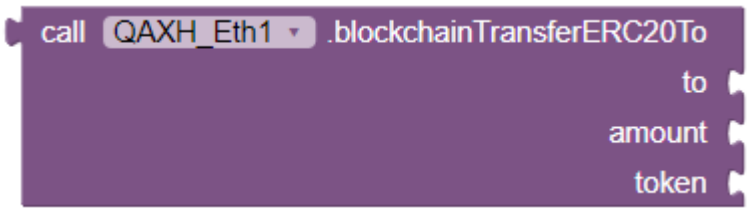
.blockchainERC20ReadVariables

Returns a list with:

1. Full name of the ERC20 token
2. Symbol name of the ERC20 token
3. Number of decimals of the token
    a. eMoney tokens are always 10 decimals

.blockchainERC20TransferTo

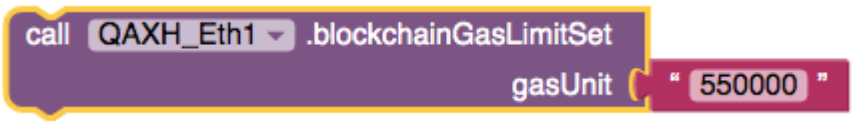
Send ERC20 token to an address



.blockchainGasLimitGet

This function returns the gasLimit in gasUnit (500 000 by default if no gasLimit Set)

.blockchainGasLimitSet



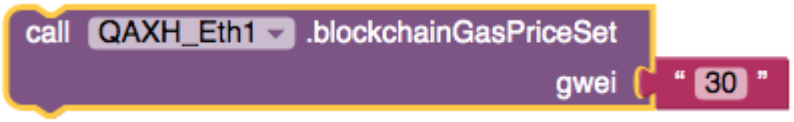
Function used to define gasLimit.

.blockchainGasPriceGet

This functions returns a list with:

- The gasPrice set in the application in GWEI
- The GasPrice read in the blockchain (in Rinkeby it is always 1) in GWEI

.blockchainGasPriceSet



This function sets the gasPrice used by the application. gasPrice in text format.

.blockchainGetVersion

Gives the version number of the qaxh_eth aix file.

The extension version number.

.blockchainKeccak

Return the keccak (SHA3 with default parameters) of the string in parameter.

.blockchainReadBalance

Gets the balance of an account.

Error : “Could not reach network”

.blockchainReadBlockNumber

Gets the current block number, ie : the last block picked up by the Infura client used in this extension.

Errors :

- “couldn reach network”
- or those returned by the web3.ethBlockNumber().send() of web3j.

.blockchainReadReceivedBlock

This function takes an address, a block number a and a number of block b. It will search for the transactions received by the address, from the a block to the (a+b -1) block (ie, during b blocks).

The return format is : c / 0x... / 0x... (as many 0x as there are received transactions), where c is the next block to read if one wants to continue reading the blockchain in the natural order (usually (a+b)).

Errors : “Problem connecting to network” or others from the web3j library.

The following example describes how to put it in a timer to read periodically the blockchain and identify all transactions sent to the addressKey parameter.

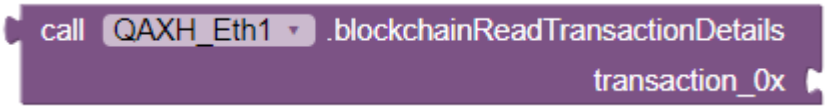
.blockchainReadTxLists

1. Address: is the Tx address to where to look at
2. startNumber: is the number of the block in the blockchain to begin with. Use a “/” to look from genesis block
3. endNumber: is the number of the last block to read. Use a “/” to look till the last mined block.

The return data is a list of lists with the following format: [receivedTx , sentTx]

When an error occurs, response data is [« couln’t get received tx » , « couldn’t get sent tx »].

. blockchainReadTransactionDetails



Retrieves all the details in a transaction.

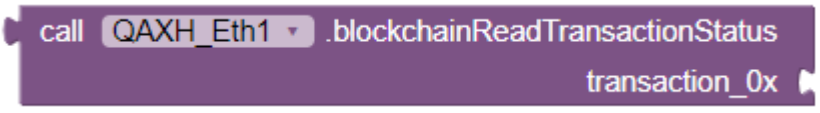
Parameters:

- String - transaction_0x, the hexadecimal id of a transaction
- Returns:
    - if the network is reachable : A list of strings containing: [Block number, From, To, Amount, Fee, Timestamp, Input data, Address of the deployed contract]
    - if not an empty list

1. 6371008 = block number
2. 0x0752763ee58a7e5f9ec3cd3038fdb70564277c46 = FROM
3. 0x3c7cecdfc3d7a607c6891b6c6ec957ed7342d400 = TO
4. 0 = amount in Ether
5. 39950000000000 = fees in WEI
6. 2020-04-24-11-29-07 = timestamp
7. 0x6f8c170a000000000000000000000000f21211e047e6dd745f359e2e0c776ab86bf5d03100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000064000000000000000000000000019f97f8439aa2b096eafb0fd974b5143342610e4 = DATA
8. *nothing* = contract address

 this function sends a runtime error of use during the mining process

. blockchainReadTransactionStatus



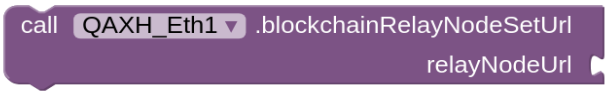
Get the status of a transaction.

- Parameters:
    - transactionId, the hexadecimal id of the transaction to scan
- Returns: a String, describing the status if successful,

- with an Error : + explanation if not.
- Values = “**Mined**”, “**Pending**”
- “Transaction has **failed** with status”
- “**failed** to pool status from transaction”
- “Error … could not **reach** network”

.blockchainReadBlockTimestamp

.blockchainRelayNodeSet



Change the Relay Node with a given Url.
Example of valid Url: (https://<**NETWORK**>.infura.io/v3/<**PROJECT_ID**>)

.blockchainRelayNodeGetVersion

Gets the client version from the Relay Node set by the .blockchainRelayNodeSet.

Errors : “Could not get version : could not reach network” or those returned by the Web3ClientVersion.getWeb3ClientVersion function of web3j

.blockchainTransferEtherTo

- Send wei (10**-18 ether)

The data field is changed to HEX in order to be read if necessary.