List of wallet Element (HYP) RPC calls:

addmultisigaddress

Syntax: addmultisigaddress <nrequired> <'["key","key"]'> [account]

addnode

Syntax: addnode <node> <add|remove|onetry>

backupwallet

Syntax: backupwallet <destination>

cccustomchange

Syntax: cccustomchange <address>

cclistcoins

Syntax: cccustomchange <address>

cclistselected

Syntax: cccustomchange <address>

ccreset

Syntax: cccustomchange <address>

ccreturnchange

Syntax: ccreturnchange <true|false>

ccselect

Syntax: ccselect <Output Hash> <Output

ccsend

Syntax: ccsend <Elementaddress> <amount>

checkwallet

Syntax: checkwallet = true / false

createrawtransaction

Syntax: createrawtransaction [{"txid":txid,"vout":n},...] {address:amount,...}

decoderawtransaction

Syntax: decoderawtransaction <hex string>

delete

Syntax: delete <address>

disablestake

This will disable staking if set true

Useful to prevent staking when diff is too high

Syntax: disablestake <true/false>

Example: disablestake true diff >5 options: diff, weight

dumpprivkey

Syntax: dumpprivkey <Elementaddress>

export difficulty

Syntax: export difficulty <interval> <directory>

getaccount

Syntax: getaccount <Elementaddress>

getaccountaddress

Syntax: getaccountaddress <account>

getaddednodeinfo

Syntax: getaddednodeinfo <dns> [node]

getaddressesbyaccount

Syntax: getaddressesbyaccount <account>

Example: getaddressesbyaccount "HYPloan"

Description: Returns the list of addresses for the given account. Surrounding the account by quote is only mandatory if there is a space in the account. If the same account has several addresses, they will all get listed.

getbalance

Syntax: getbalance [account] [minconf=1]

getblock

Syntax: getblock <hash> [txinfo]

getblock

Syntax: getblock <number> [txinfo]

getblockcount

Syntax: getblockcount

getblockhash

Syntax: getblockhash <index>

getblocktemplate

Syntax: getblocktemplate [params]

getcheckpoint

Syntax: getcheckpoint

getconnectioncount

Returns the number of connections to other nodes.

This command has no argument. It gives the same result as looking at one of the popup in the lower right corner.

getdifficulty

Returns the difficulty as a multiple of the minimum difficulty.

This command has no argument. You can get the same information on the GUI by looking at a popup on the lower right. Still, getdifficulty will also tell you the proof-of-stake difficulty (even when the proof-of-stake period is over)

getgenerate

Returns true or false.

This command has no argument. Its purpose is uncertain.

getinfo

Returns an object containing various state info.

This command gives you a lot of useful info, especially when debugging. It has no parameter.

getmininginfo

Returns an object containing mining-related information.

This command has no argument. The GUI can give the same information by looking at one of the popup in the lower right corner.

getmoneysupply

Syntax: getmoneysupply [height]

getnewaddress

Syntax: getnewaddress [account]

getnewpubkey

Syntax: getnewpubkey [account]

getpeerinfo

Shows how many peers your wallet has connection to.

getrawmempool

Syntax: getrawmempool

getrawtransaction

Syntax: getrawtransaction <txid> [verbose=0]

getreceivedbyaccount

Syntax: getreceivedbyaccount <account> [minconf=1]

getreceivedbyaddress

Syntax: getreceivedbyaddress <Elementaddress> [minconf=1]

getstakesplitthreshold

Syntax:

getstaketx

Syntax: getstaketx <txid>

gettransaction

Syntax: gettransaction <txid>

getwork

Syntax: getwork [data]

getworkex

Syntax: getworkex [data, coinbase]

help

Syntax: help [command]

importprivkey

Syntax: importprivkey <Elementprivkey> [label]

keypoolrefill

Syntax:

listaccounts

Syntax: listaccounts [minconf=1]

listaddressgroupings

Syntax:

listreceivedbyaccount

Syntax: listreceivedbyaccount [minconf=1] [includeempty=false]

listreceivedbyaddress

Syntax: listreceivedbyaddress [minconf=1] [includeempty=false]

listsinceblock

Syntax: listsinceblock [blockhash] [target-confirmations]

listtransactions

Syntax: listtransactions [account] [count=10] [from=0]

listunspent

Syntax: listunspent [minconf=1] [maxconf=9999999] ["address",...]

makekeypair

Syntax: makekeypair [prefix]

moneysupply

Syntax:

move

Syntax: move <fromaccount> <toaccount> <amount> [minconf=1] [comment]

repairwallet

Syntax:

rescanfromblock

Syntax: rescanfromblock <block height>

resendtx

Syntax:

reservebalance

Syntax: reservebalance [<reserve> [amount]]

sendalert

Syntax: sendalert <message> <privatekey> <minver> <maxver> <priority> <id> [cancelupto]

sendfrom

Syntax: sendfrom <fromaccount> <toElementaddress> <amount> [minconf=1] [comment] [comment-to]

sendmany

Syntax: sendmany <fromaccount> {address:amount,...} [minconf=1] [comment]

sendrawtransaction

Syntax: sendrawtransaction <hex string>

sendtoaddress

Syntax: sendtoaddress <Elementaddress> <amount> [comment] [comment-to]

setaccount

Syntax: setaccount <Elementaddress> <account>

setstakesplitthreshold

Syntax: setstakesplitthreshold <1 - 999,999>

Description: A stake output will split (in two parts), when the resulting amount for each of the outputs equals or exceeds the threshold value specified. For example: by setting a threshold of 5,000, stake outputs greater than 10,000 will be split into two.

settxfee

Syntax: settxfee <amount>

signmessage

Syntax: signmessage <Elementaddress> <message>

signrawtransaction

Syntax: signrawtransaction <hex string> [{"txid":txid,"vout":n,"scriptPubKey":hex},...] [<privatekey1>,...] [sighashtype="ALL"]

stakeforcharity

Syntax: stakeforcharity <Element Address> <percent> [Change Address] [min amount] [max amount]

strictprotocol

Syntax:

strictincoming

Syntax:

stop

Syntax:stop <detach>

submitblock

Syntax: submitblock <hex data> [optional-params-obj]

validateaddress

Syntax: validateaddress <Elementaddress>

validatepubkey

Syntax: validatepubkey <Elementpubkey>

verifymessage

Syntax: verifymessage <Elementaddress> <signature> <message>

walletlock

Removes the wallet encryption key from memory, locking the wallet. After calling this method, you will need to call walletpassphrase again before being able to call any methods which require the wallet to be unlocked.

This command has no parameter. Once you used it, you unlock the wallet with walletpassphrase — see below for details

walletpassphrase

Stores the wallet decryption key in memory for <timeout> seconds. mintonly is optional true/false allowing only block minting.

walletpassphrase <passphrase> <timeout> [mintonly]

So walletpassphrase <passphrase> 3600 true means "unlock for pos for one hour". Note that the GUI allows for a permanent unlock for pos, but the RPC requires a timeout; that being said, a very long timeout like 999999999 is akin to permanent. You cannot move from true to false directly (from "unlock for pos" to "unlock"), you must use walletlock in the middle. So to set to unlock for pos then to unlock, you must use these three commands

walletpassphrase <passphrase> 3600 true

walletpasslock

walletpassphrase <passphrase> 3600 false

walletpassphrasechange

Syntax: walletpassphrasechange <oldpassphrase> <newpassphrase>