Wallet

WalletBalance

WalletBalance returns the balance of the given address at the current head of the chain.

Perms: read

Inputs:

```

Copy ["f01234"]

```

Response:"0"

WalletDefaultAddress

WalletDefaultAddress returns the address marked as default in the wallet.

Perms: write

Inputs:null

Response:"f01234"

WalletDelete

WalletDelete deletes an address from the wallet.

Perms: admin

Inputs:

```

Copy ["f01234"]

```

Response:{}

WalletExport

WalletExport returns the private key of an address in the wallet.

Perms: admin

Inputs:

```

Copy ["f01234"]

```

Response:

```

Copy { "Type":"bls", "PrivateKey":"Ynl0ZSBhcnJheQ==" }

```

WalletHas

WalletHas indicates whether the given address is in the wallet.

Perms: write

Inputs:

```

Copy ["f01234"]

```

Response:true

WalletImport

WalletImport receives a KeyInfo, which includes a private key, and imports it into the wallet.

Perms: admin

Inputs:

```

Copy [ { "Type":"bls", "PrivateKey":"Ynl0ZSBhcnJheQ==" } ]

```

Response:"f01234"

WalletList

WalletList lists all the addresses in the wallet.

Perms: write

Inputs:null

Response:

```

Copy ["f01234"]

```

WalletNew

WalletNew creates a new address in the wallet with the given sigType. Available key types: bls, secp256k1, secp256k1-ledger Support for numerical types: 1 - secp256k1, 2 - BLS is deprecated

Perms: write

Inputs:

```

Copy ["bls"]

```

Response:"f01234"

WalletSetDefault

WalletSetDefault marks the given address as as the default one.

Perms: write

Inputs:

```

Copy ["f01234"]

```

Response:{}

WalletSign

WalletSign signs the given bytes using the given address.

Perms: sign

Inputs:

```

Copy ["f01234","Ynl0ZSBhcnJheQ=="]

```

Response:

```

Copy { "Type":2, "Data":"Ynl0ZSBhcnJheQ==" }

```

WalletSignMessage

WalletSignMessage signs the given message using the given address.

Perms: sign

Inputs:

```

Copy [ "f01234", { "Version":42, "To":"f01234", "From":"f01234", "Nonce":42, "Value":"0", "GasLimit":9, "GasFeeCap":"0",
"GasPremium":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "CID":{
"/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } } ]

```

Response:

```

Copy { "Message":{ "Version":42, "To":"f01234", "From":"f01234", "Nonce":42, "Value":"0", "GasLimit":9, "GasFeeCap":"0",
"GasPremium":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "CID":{
"/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } }, "Signature":{ "Type":2,
"Data":"Ynl0ZSBhcnJheQ==" }, "CID":{ "/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } }

```

WalletValidateAddress

WalletValidateAddress validates whether a given string can be decoded as a well-formed address

Perms: read

Inputs:

```

Copy ["string value"]

```

Response:"f01234"

WalletVerify

WalletVerify takes an address, a signature, and some bytes, and indicates whether the signature is valid. The address does not have to be in the wallet.

Perms: read

Inputs:

```

Copy [ "f01234", "Ynl0ZSBhcnJheQ==", { "Type":2, "Data":"Ynl0ZSBhcnJheQ==" } ]

```

Response:true

Previous Sync Next Web3

Last updated5 months ago