Gate Wallet Integration Doc

EVM Chain Method

Global variable

Gatewallet inherits ethereum variables. When you use Gatewallet variables, you need to modify the connector based on your own projectwindow.ethereum.

- window.ethereum
- window.gatewallet

Check whether Gate Wallet is installed

```
if (typeof window.gatewallet !== 'undefined') {
  console.log('Gate Wallet is installed!');
}
```

`eth_requestAccounts` Connect account

```
const newAccounts = await gatewallet.request({ method: 'eth_requestAccounts' })
    .then(handleAccountsChanged)
    .catch((error) => {
      if (error.code === 4001) {
        // EIP-1193 userRejectedRequest error
        console.log('Please connect to Gate Wallet.');
      } else {
        console.error(error);
      }
});
```

`eth_accounts` Get account

```
const _accounts = await gatewallet.request({ method: 'eth_accounts' });

console.log(_accounts) //0x...11
console.log(window.gatewallet.selectedAddress) //0x...11
```

`eth_sign` Signature

```
const accounts = '0x09fb420ed4059d3e67da7a53e38a8fe7fa12a8c2'
const msg = '0x879a053d4800c6354e76c7985a865d2922c82fb5b3f4577b2fe08b998954f2e0'
const ethResult = await gatewallet.request({
    method: 'eth_sign',
    params: [accounts, msg],
});
```

`personal_sign` Authorized Signature

```
const from = '0x09fb420ed4059d3e67da7a53e38a8fe7fa12a8c2'
const msg = `消息体`;
const sign = await gatewallet.request({
    method: 'personal_sign',
    params: [msg, from, 'Example password'],
});

console.log(sign);  //0x650e...
```

`eth_sendTransaction` Transaction

```
const transactionParameters = {
  nonce: '0x00',
  gasPrice: '0x09184e72a000',
  gas: '0x2710',
  to: '0x0000000000000000000000000000000000000000',
  from: gatewallet.selectedAddress,
  value: '0x00',
  data:
    '0x7f7465737432000000000000000000000000000000000000000000000000000000600057'
  chainId: '0x3',
};

const txHash = await gatewallet.request({
   method: 'eth_sendTransaction',
   params: [transactionParameters],
});
```

Sign Typed Data

`eth_signTypedData_v3`

`eth_signTypedData_v4`

```

const sign = await gatewallet.request({
    method: 'eth_signTypedData_v3',
    params: [from, JSON.stringify(msgParams)],
});

const sign = await gatewallet.request({
    method: 'eth_signTypedData_v4',
    params: [from, JSON.stringify(msgParams)],
});
```

accountsChange

```
gatewallet.on('accountsChange',(res)=>{
    console.log(res)
})
```

`gateAccountsChange`

This method is used when the wallet account (gate Accountinfo)is switched.

Only Gate Wallet has this method injected.

```
window?.gatewallet?.on('gateAccountChange', fun)

return () => {
  window?.gatewallet?.removeListener('gateAccountChange', fun)
}
```

Bitcoin Chain

Transaction

`window.gatewallet.bitcoin.sendBitcoin({ fromAddress, toAddress, satoshis, options })`

Parameter

- fromAddress - string: Transfer address
- toAddress - string: Receiving address
- satoshis - string: Quantity originated (satoshis)
- options - object: （Optional）Custom rate
  - feeRate - number：Network rate

Returned value

- Promise - object
  - txhash

demo

```javascript
try {
   await window.gatewallet.bitcoin.sendBitcion({ fromAddress, toAddress,
satoshis, options })
} catch (err) {
    console.error(err)
}
```

signMessage

`window.gatewallet.bitcoin.signMessage({ fromAddress, text, type })`

Parameter

- fromAddress - string: Signed address
- text - string: Signed message
- type - "ecdsa" | "bip322-simple": （Optional）The default is "ecdsa"

Returned value

- Promise - string
  - signature: Signature result

Inscription

`window.gatewallet.bitcoin.inscribeTransfer({ fromAddress, toAddress, ticker, amount, isOrdinals })`

Parameter

- fromAddress - string: Inscribed address
- toAddress - string: Address of receive
- ticker - string: Inscription tick
- amount - string: （Optional）Quantity of inscriptions
- isOrdinals - boolean: Is it Ordinals deal

Returned value

- Promise - object
  - [hash1，hash2]: commit and review`s hash of the broadcast

Send inscription

`window.gatewallet.bitcoin.sendInscription({ fromAddress, toAddress, inscriptionId, options })`

Parameter

- fromAddress - string: Address of send
- toAddress - string: Address of receive
- inscriptionId - string: Inscription id
- amount - string: Transfer quantity
- options - object: （Optional）Custom rate
  - feeRate - number：Network rate
  - amount - number: Quantity

Returned value

- Promise - object
  - hash: Broadcast hash

Deploy

`window.gatewallet.bitcoin.deploy({fromAddress, toAddress, ticker, max, limit, decimals})`

Parameter

- fromAddress - string: Address of send
- toAddress - string: Address of receive
- ticker - string: Inscription name
- max - string: Total supply
- limit - string: Mint limits the number of times in a single session
- decimals - string: （Optional）decimals

Returned value

- Promise - object

  {

  - "hash_raw_tx_1" - string
  - "hash_raw_tx_2" - string

  }

Mint

`window.gatewallet.bitcoin.mint({fromAddress, toAddress, ticker, amount, repeat})`

Parameter

- fromAddress - string: Address of send
- toAddress - string: Address of receive
- ticker - string: Inscription name
- amount - string: mint quantity
- repeat - string: Times of repetition

Returned value

- Promise - object

  {

  - "hash_raw_tx_1" - string
  - "hash_raw_tx_2" - string

  }

Inscription NFT

`window.gatewallet.bitcoin.inscribeNFT({fromAddress, toAddress, mimeType, payload})`

Parameter

- fromAddress - string: Address of send
- toAddress - string: Address of receive
- mimeType - string: Type of inscription （image/png、image/jpeg、image/svg+xml、text/html、text/plain、text/csv etc.）
- payload - array[string]: An array of hexadecimal images, text, etc.

signPsbt （Put on shelves, buy, etc）

`window.gatewallet.bitcoin.signPsbt({fromAddress, psbtHex, options})`

Parameter

- fromAddress - string: BTC address to be signed
- psbtHex - string:
- options - object （Optional）
  - `autoFinalized` - `boolean` : whether finalize psbt after signing, default is true
  - `toSignInputs` - `array` :
    - `index` - `number` : which input to sign
    - `address` - `string` : (at least specify either an address or a publicKey) Which corresponding private key to use for signing
    - `publicKey` - `string` : (at least specify either an address or a publicKey) Which corresponding private key to use for signing
    - `sighashTypes` - `number[]` : (optionals) sighashTypes
    - `disableTweakSigner` - `boolean` :(optionals) When signing and unlocking Taproot addresses, the `tweakSigner` is used by default for signature generation. Enabling this allows for signing with the original private key.

signPsbts （signPsbt batch）

`window.gatewallet.bitcoin.signPsbts({fromAddress, psbtHexs, options})`

Parameter

fromAddress - string: BTC address to be signed `psbtHexs` - `string[]`： psbtHex array

`options` - `object` []: （Optional)

- `autoFinalized` - `boolean` : whether finalize psbt after signing, default is true
- `toSignInputs` - `array` :
    - `index` - `number` : which input to sign
    - `address` - `string` : (at least specify either an address or a publicKey) Which corresponding private key to use for signing
    - `publicKey` - `string` : (at least specify either an address or a publicKey) Which corresponding private key to use for signing
    - `sighashTypes` - `number[]` : (optionals) sighashTypes

gateAccountInfo

Variable acquisition

`window?.gatewallet?.gateAccountInfo`

Field details

`walletName` - `string`： Wallet name

`accountName` - `string`： Account name

`walletId` - `string`： Unique id

`accountNetworkArr` - `array` : [{

`accountFormat` - `string`： Account format

`accountFormatName` - `string`： The account format corresponds to the full name

`address` - `string`： Account address

`network` - `string`：Chain abbreviation

`accountPublicKey` - `string`： （Optional）Account public key（Only Bitcoin chain has this field）

}],

`moreAddressSort` - `array`： Multiple address sort

gateAccountChange Event

Event listening

```javascript
window?.gatewallet?.on('gateAccountChange', (gateAccountInfo) => {
    console.log('---gateAccountInfo---', gateAccountInfo)
}))
```

Event removal

```javascript
window?.gatewallet?.removeListener('gateAccountChange', function)
```