“The Cold Wallet 2.0”

Open Source

Web App demo & JavaScript Lib

V1.7.4

©2018 ELLIPAL

Background

ELLIPAL's secure hardware wallet keeps the user's private keys isolated from any kind of wired and wireless connections to outside of the wallet. The hardware is a signing device. The only way to exchange data of tx to be signed and signed is QR code scanning, which is open, visual and participated by the user.

To enable 3rd parties to integrate ELLIPAL's hardware to provide more services to end users, ELLIPAL developed open source JavaScript lib which could be used to deal with full functions of account management, balance check and sending coin or tokens. The lib and demo work in web browsers on PC(Chrome), Mac(Safari), Android(Chrome) and iPhone(Safari). The lib can also be integrated into web-based service or mobile App via web view or nodejs.

This document shows the interfaces between web demo app and JavaSciprt.

The information provided in this document is intended for informational purposes only and is subject to change without notice. Please always check www.ellipal.com or GitHub https://github.com/ellipal to get the latest information.

©2018 ELLIPAL

General steps

1. An apiKey is needed to access ELLIPAL services to check balance and transfer coins.
   Please visit http://wallet.ellipal.com/key-reg to get the apiKey.
2. Demo App Import JQ and ellipal.js
3. Call the suitable interface the ellipal.js

©2018 ELLIPAL

Interfaces

API	Function	Remark
1. ellipal.getApiVersion	Get current API version	
2. ellipal.connectWallet	Connect hardware wallet to get account address of particular coin or token	
3. ellipal.getBalance	Get balance of an address	
4. ellipal.prepareTx	Prepare data to be signed of a transaction	Used to send coin or token to others.
5. ellipal.getSignedTx	Decode signed QR code data from hardware wallet.	
6. ellipal.broadcastTx	Send signed transaction to blockchain p2p networks	
7. ellipal. getAverageMinerFee	Get current average miner fee/gas	End user may change miner fee to shorten mined time on blockchain.

ellipal.getApiVersion( )

Get current API version.

ellipal.connectWallet(syncUrl)

Connect hardware wallet to get account address of particular coin or token.

Input

Item	Mandatory	Remark
syncUrl	yes	String decoded from account QR code

Result

Item	Sample	Remark

©2018 ELLIPAL

Item	Sample	Remark
name	Ellipal-btc	Account name
type	“BTC” “ETH”	Coin group type. ERC20 tokens are under ETH
addr		Account address
pubkey		Pubkey(BTC/BCH only)
legacyAddr		LegacyAddress(BCH only)

ellipal.getBalance(apiKey, cType, addr)

Get balance of an address.

Input

Item	Mandatory	Remark
apiKey	Yes	apiKey from ELLIPAL
cType	Yes	Coin group type. ERC20 tokens are under ETH
addr	Yes	Account address

Result

Item	Sample	Remark
status	true	Flag of success
code	10000	Error code
msg	“Please try again later”	Error msg/ success
data	{ }	data

©2018 ELLIPAL

Data details

Item	Sample	Remark
balanceTotalUSD	“1.49”	Total estimated value in USD
dataList	[{ }]	JSON format balance. For ETH type address, there will be some ERC20 tokens under this address

dataList

Item	Sample	Remark
address		Account address
contractAddr		Smart contract(only for Ethereum ERC20)
cType	“BTC”	Coin group type
balance		balance
decimal		Decimal number
logo		Logo of coin or token
symbol	“BNB”	ERC20 symbol
usdValue	“0.63”	Estimated value in USD

ellipal.prepareTx(apiKey, paramObj)

Prepare data to be signed of a transaction.

Input

Item	Mandatory	Remark
apiKey	Yes	apiKey from ELLIPAL

©2018 ELLIPAL

Item	Mandatory	Remark
paramObj	Yes	Parameters object

paramObj

Key	Value	Mandatory	Sample	Remark
httpType	req	Yes	“req”	action
cType	ETH/BTC	Yes	“BTC”	Coin group type
fee		Yes		Miner fee(ETH and ERC20 is gas price)
cAddress		Yes		Sender address
dAddress		Yes		Receiver address
contractAddr		ERC		ERC20 contract address
amount		Yes	100000	Sending amount
remark		No		remark

Result

Item	Sample	Remark
status	true	Status of success
code	0	Error code
msg	“success”	Error message / success
data	{ }	data

data

Item	Sample	Remark
txDigest		Tx digest

©2018 ELLIPAL

Item	Sample	Remark
tx		Tx data to be signed

ellipal.getSignedTx (signUrl)

Decode signed QR code data from hardware wallet.

Input

Item	Mandatory	Remark
signUrl	Yes	Cold wallet returned qrcode data

Result

Item	Sample	Remark
curr	2	Current index
total	8	Total segments
type	“BTC”	Coin group type
addr		Account address
signedData		Signed data of this segment

ellipal.broadcastTx(apiKey, paramObj)

Send signed transaction to blockchain p2p networks.

Input

Item	Mandatory	Remark

©2018 ELLIPAL

Item	Mandatory	Remark
apiKey	Yes	apiKey from ELLIPAL
paramObj	Yes	Parameters object

paramObj

Key	Value	Mandatory	Sample	Remark
httpType	send	Yes	“send”	Action
cType		Yes	“BTC”	Coin group type “ERC” for all ERC20 tokens
fee		Yes		Miner fee or gas price(ETH)
cAddress		Yes		Sender address
dAddress		Yes		Receiver address
contractAddr		ERC20 Yes		ERC20 smart contract address
amount		Yes	100000	Sending amount
r	“oxabcdefg”	ETH/ERC		ETH/ERC20 signed data from hardware wallet
s	“oxabcdefg”	ETH/ERC		ETH/ERC20 signed data from hardware wallet
v	“oxabcdefg”	ETH/ERC		ETH/ERC20 signed data from hardware wallet
sig		BTC/BCH		BTC/BCH signed data from hardware wallet
tx_unsigned		BTC/BCH		Unsigned Tx from “req”
pubkey		BTC/BCH		
remark		No		Remark

©2018 ELLIPAL

Result

Item	Sample	Remark
status	true	Status of success
code	0	Error code
msg	“success”	Error message / success

ellipal.getAverageMinerFee (apiKey, cType)

Get current average miner fee/gas.

Input

Item	Mandatory	Remark
apiKey	Yes	apiKey from ELLIPAL
cType	Yes	Coin group type. ERC20 tokens are under same address of ETH

Result

Item	Sample	Remark
code	0	Error code
msg	“cType error”	Error message / success
status	true	Status of success
data	{ }	data

data

Item	Sample	Remark

©2018 ELLIPAL

Item	Sample	Remark
gasMul	1	Estimated steps of gas consuming (Only for ETH and ERC20 Tokens)
gasPrice	2400	Miner fee or gas price
decimal	8	Decimal number
gasUnit	“BTC”	Miner fee type

Coin group type

In the current version of ELLIPAL’s design, a coin group refers to a blockchain like Bitcoin, Bitcoin Cash or Ethereum. ERC is used as coin group type only for sending a particular ERC20 token. When sending an ERC20 token, the token was distinguished by its smart contract address as there are many duplicate token names or symbols.

©2018 ELLIPAL