bitcoinjs-message

TS

2.2.0 • Public • Published 3 years ago

bitcoinjs-message

npm v2.2.0 build no longer available Dependency status

JS Standard Code Style

Examples (Note about Electrum support at the bottom)

```
var bitcoin = require('bitcoinjs-lib') // v4.x.x
var bitcoinMessage = require('bitcoinjs-message')
```

> sign(message, privateKey, compressed[, network.messagePrefix, sigOptions])
>
> - If you pass the sigOptions arg instead of messagePrefix it will dynamically replace.

> - sigOptions contains two attributes
> - `segwitType` should be one of `'p2sh(p2wpkh)'` or `'p2wpkh'`
> - `extraEntropy` will be used to create non-deterministic signatures using the RFC6979 extra entropy parameter. R value reuse is not an issue.

Sign a Bitcoin message

```
var keyPair = bitcoin.ECPair.fromWIF('L4rK1yDtCWekvXuE6oXD9jCYfF
var privateKey = keyPair.privateKey
var message = 'This is an example of a signed message.'

var signature = bitcoinMessage.sign(message, privateKey, keyPair
console.log(signature.toString('base64'))
// => 'H9L5yLFjti0QTHhPyFrZCT1V/MMnBtXKmoiKDZ78NDBjERki6ZTQZdSMC
```

To produce non-deterministic signatures you can pass an extra option to sign()

```
var { randomBytes } = require('crypto')
var keyPair = bitcoin.ECPair.fromWIF('L4rK1yDtCWekvXuE6oXD9jCYfF
var privateKey = keyPair.privateKey
var message = 'This is an example of a signed message.'

var signature = bitcoinMessage.sign(message, privateKey, keyPair
console.log(signature.toString('base64'))
// => different (but valid) signature each time
```

Sign a Bitcoin message (with segwit addresses)

```
// P2SH(P2WPKH) address 'p2sh(p2wpkh)'
var signature = bitcoinMessage.sign(message, privateKey, keyPair
console.log(signature.toString('base64'))
// => 'I9L5yLFjti0QTHhPyFrZCT1V/MMnBtXKmoiKDZ78NDBjERki6ZTQZdSMC
```

```
// P2WPKH address 'p2wpkh'
var signature = bitcoinMessage.sign(message, privateKey, keyPair
console.log(signature.toString('base64'))
// => 'J9L5yLFjtiOQTHhPyFrZCT1V/MMnBtXKmoiKDZ78NDBjERki6ZTQZdSMC
```

> verify(message, address, signature[, network.messagePrefix, checkSegwitAlways])

Verify a Bitcoin message

```
var address = '1F3sAm6ZtwLAUnj7d38pGFxtP3RVEvtsbV'

console.log(bitcoinMessage.verify(message, address, signature))
// => true
```

About Electrum segwit signature support

- For Signing: Use the non-segwit compressed signing parameters for both segwit types (p2sh-p2wpkh and p2wpkh)
- For Verifying: Pass the checkSegwitAlways argument as true. (messagePrefix should be set to null to default to Bitcoin messagePrefix)

LICENSE MIT

Keywords

bitcoinjs-message bitcoin

Install

```
> npm i bitcoinjs-message
```

Repository

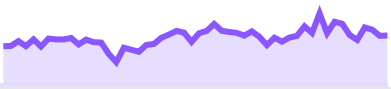
github.com/bitcoinjs/bitcoinjs-message

Homepage

github.com/bitcoinjs/bitcoinjs-message

Weekly Downloads

31,713



Version	License
2.2.0	MIT

Unpacked Size	Total Files
13.7 kB	6

Issues	Pull Requests
2	1

Last publish

3 years ago

Collaborators



>_Try on RunKit

Report malware

Support

Help

Advisories

Status

Contact npm

Company

About

Blog

Press

Terms & Policies

Policies

Terms of Use

Code of Conduct

Privacy