OBI Module

Oracle Binary Encoding (OBI) is the standard way to serialized and deserialize binary data in the BandChain ecosystem. This module provides the functionality to serialize data.More details .

Bandchain.js provide a class namedObi to help encode/decode binary data using OBI encoding. Here is the usage of the class.

Constructor

- schema
- string
-
    - A string of OBI schema, including input and output schemas.

Example

import

{

Obi

}

from

'@bandprotocol/bandchain.js'

const obi =

new

Obi ('{symbol:string, px: u64, w: {a: u8, b: u8}, tb: [string]} / string')

encodeInput(value)

Encode the value based on given OBI input schema

Parameter

- valueany
-
    - A value to be encoded. can be any type of data.

Return

- Buffer
-
    - An encoded value

Example

import

{

Obi

}

from

'@bandprotocol/bandchain.js'

const obi =

new

Obi ('{symbol:string, px: u64, w: {a: u8, b: u8}, tb: [string]} / string') const testInput =

{ symbol :

'BTC' , px :

9000 , w :

{

a :

1 ,

b :

2

} , tb :

['a' ,

'b'] , } console . log (obi . encodeInput (testInput) . toString ('hex')) Result

0000000342544300000000000023280102000000020000000161000000016 2

encodeOutput(value)

Encode the value based on OBI output schema

Parameter

- valueany
-
    - The value to be encoded

Return

- Buffer
-
    - An encoded value

Example

import

{

Obi

}

from

'@bandprotocol/bandchain.js'

const obi =

new

Obi ('{symbol:string, px: u64, w: {a: u8, b: u8}, tb: [string]} / string') const testOutput =

'test' console . log (obi . encodeOutput (testOutput) . toString ('hex')) Result

0000000474657374

decodeInput(buff)

Decode the value based on given OBI input schema

Parameter

- valueBuffer
-

- The value to be decoded

Return

- any
-
  - A decoded value

Exceptions

Type Description DecodeError Not all data is consumed after decoding output Example

import

{

Obi

}

from

'@bandprotocol/bandchain.js'

const obi =

new

Obi ('{symbol:string, px: u64, w: {a: u8, b: u8}, tb: [string]} / string') console . log (obi . decodeInput (Buffer . from ('000000034254430000000000002328010200000020000000161000000162' ,

'hex'))) Result

{

"symbol" :

"BTC" ,

"px" :

9000n ,

"w" :

{

"a" :

1n ,

"b" :

2n

} ,

"tb" :

["a" ,

"b"]

}

decodeOutput(buff)

Decode the output value by using output schema

Parameter

- valueBuffer
-

- The value to be decoded

Return

- any
-
  - A decoded value

Exceptions

Type Description DecodeError Not all data is consumed after decoding output Example

import

{

Obi

}

from

'@bandprotocol/bandchain.js'

const obi =

new

Obi ('{symbol:string, px: u64, w: {a: u8, b: u8}, tb: [string]} / string') console . log (obi . decodeOutput (Buffer . from ('0000000474657374' ,

'hex'))) Result

test Previous Message Module Next Transaction Module