

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](#)

Constructor

- schema
-
- : Input and output schema.

Example

```
from pyband import PyObi
```

obi

```
PyObi ( "{symbol: string,px: u64,in: {a: u8,b: u8}, tb:bool} / string" )
```

encode_input(value)

This function encodes the input value by using input schema.

Parameter

- value
-
- : Value to be encoded.

Return

-
- : An encoded value

Example

```
from pyband . obi import PyObi
```

obi

```
PyObi ( "{symbol: string,px: u64,in: {a: u8,b: u8}, tb:bool} / string" ) test_input =
```

```
{ "symbol" :
```

```
"BTC" ,
```

```
"px" :
```

```
9000 ,
```

```
"in" :
```

```
{ "a" :
```

```
1 ,
```

```
"b" :
```

```
2 } ,
```

```
"tb" :
```

```
False }
```

```
print ( obi . encode_input ( test_input ) . hex ( ) ) Result
```

```
00000003425443000000000000002328010200
```

encode_output(value)

This function encodes the output value by using output schema.

Parameter

- value
-
- : Value to be encoded.

Return

-
- : An encoded value

Example

```
from pyband . obi import PyObi
```

obi

```
PyObi ( "{symbol: string,px: u64,in: {a: u8,b: u8}, tb:bool} / string" ) test_output =
```

```
"test" print ( obi . encode_output ( test_output ) . hex ( ) ) Result
```

```
0000000474657374
```

decode_input(value)

This function decode the input value by using input schema

Parameter

- value
-
- : Value to be decoded.

Return

-
- : A decoded value

Example

```
from pyband . obi import PyObi
```

obi

```
PyObi ( "{symbol: string,px: u64,in: {a: u8,b: u8}, tb:bool} / string" ) print ( obi . decode_input ( bytearray . fromhex ( "0000000342544300000000000002328010200" ) ) ) Result
```

```
{"symbol": "BTC", "px": 9000, "in": {"a": 1, "b": 2}, "tb": False}
```

decode_output(value)

This function decode the output value by using output schema

Parameter

- value
-
- : Value to be decoded.

Return

-
- : A decoded value

Example

```
from pyband . obi import PyObi
```

obi

```
PyObi ( "{symbol: string,px: u64,in: {a: u8,b: u8}, tb:bool} / string" ) print ( obi . decode_output ( bytearray . fromhex (
"0000000474657374" ) ) ) Result
```

test [Previous Data Module](#) [Next Transaction Module](#)