Ethereum ABI Utils Documentation

Release 0.1.0

Piper Merriam <pipermerriam@gmail.com>

Oct 27, 2017

Contents

1 Overview **3**
1.1 Credit 3

2 Encoding **5**

3 Decoding **7**

4 Indices and tables **9**

Contents:

CHAPTER 1

Overview

This library provides low level utilities for ABI encoding and decoding.

Credit

This code was extracted from the `pyethereum` library authored by Vitalik Buterin.

CHAPTER 2

Encoding

These functions are intended for encoding python values into representations that are suitable for interacting with the EVM.

- `eth_alarm.encode_single(type, value)`

This function encodes `value` in the ABI encoding for the provided `type`.

```
>>> encode_single('uint256', 12345)

↪'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
↪'
```

The **value** parameter is expected to be one of the recognized EVM types.

Note: This function cannot be used to encode array types such as `bytes32[]`.

- `eth_alarm.encode_abi(types, values)`

This function encodes `values` in the ABI encoding for the corresponding type provided by the `types` argument.

```
>>> encode_abi(['uint256'], [12345])

↪'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
↪'
>>> encode_abi(['bytes32', 'bytes32'], ['a', 'b'])

↪'a\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
↪'
```

The **values** parameter is expected to be an iterable whose values are all one of the recognized EVM types.

CHAPTER 3

Decoding

These functions are intended for decoding return values from the EVM.

- `eth_alarm.decode_single(type, data)`

This function tries to decode `data` into the python type that corresponds to the provided `type`. This function accepts both byte strings as well as their hexidecimal representation with or without the `0x` prefix.

```
>>> decode_single('uint256',
↪'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
↪')
12345
>>> decode_single('uint256',
↪'0000000000000000000000000000000000000000000000000000000000003039')
12345
>>> decode_single('uint256',
↪'0x0000000000000000000000000000000000000000000000000000000000003039')
12345
```

The **value** parameter is expected to be one of the recognized EVM types.

Note: This function cannot be used to decode dynamic or array types such as `bytes32[]`.

- `eth_alarm.decode_abi(types, data)`

This function decodes `data` into the python type corresponding to the provided `types`. This function accepts both byte arrays as well as their hexidecimal representation with or without the `0x` prefix.

```
>>> decode_abi(['uint256'],
↪'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
↪')
[12345]
>>> decode_abi(['bytes32', 'bytes32'],
↪'a\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
↪')
```

```
['a', 'b']
```

The **values** parameter is expected to be an iterable whose values are all one of the recognized EVM types.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search