How to Parse Ethereum Token Transfer Events

API Link

https://api.etherscan.io/api?module=logs&action=getLogs&fromBlock=3957992&toBlock=100000000&address=0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0&topic0=0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef&apikey=YOURKEY

There are several parameters:

Prefix: https://api.etherscan.io/api?module=logs&action=getLogs

Block Range: fromBlock=3957992&toBlock=100000000

(Token) Contract Address: address=0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0

Transfer Topic Hash: topic0=0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef

My API Key: apikey=YOURKEY

Since prefix, block range, contract address and API key are straight forward, I am going to emphasize on Transfer Topic Hash. But to get into Transfer Topic Hash, first we need to understand the content of Ethereum events.

Ethereum Events

The following is an example of Ethereum event log.


```
{"address":"0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0","topics":
["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef","0x000000000000
000000000000d0a6e6c54dbc68db5db3a091b171a77407ff7ccf","0x0000000000000000000000000ba80
9aeb7f910d44ece7f1f17af99b489d010ffc"],"data":"0x0000000000000000000000000000000000000
00000002143420340dabfb0de","blockNumber":"0x3c64e8","timeStamp":"0x5957a583","gasP
rice":"0x4e3b29200","gasUsed":"0x143ae","logIndex":"0x23","transactionHash":"0x081357
95a5c917b6eff494f5778f6b33e810fe6ed3646799a36d835aa3f94c64","transactionIndex":"0x1b"
}
```


There are three topics separated by commas:


```
"topics":
["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef","0x000000000000
000000000000d0a6e6c54dbc68db5db3a091b171a77407ff7ccf","0x0000000000000000000000000ba80
9aeb7f910d44ece7f1f17af99b489d010ffc"]
```


The first topic is the transfer signature (aka. Transfer Topic Hash), the second and third topics are the *from* and *to* addresses. The *from* address is the address of the sender, while the *to* address is the address of the receiver.

Ethereum Events

Canonical Signature


```
{"address":"0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0","topics":
["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef","0x000000000000
0000000000000d0a6e6c54dbc68db5db3a091b171a77407ff7ccf","0x0000000000000000000000000ba80
9aeb7f910d44ece7f1f17af99b489d010ffc"],"data":"0x0000000000000000000000000000000000000
00000002143420340dabfb0de","blockNumber":"0x3c64e8","timeStamp":"0x5957a583","gasP
rice":"0x4e3b29200","gasUsed":"0x143ae","logIndex":"0x23","transactionHash":"0x081357
95a5c917b6eff494f5778f6b33e810fe6ed3646799a36d835aa3f94c64","transactionIndex":"0x1b"
}
```


If you look at the ERC20 standard, you will find two event functions:

```
event Transfer(address indexed _from, address indexed _to, uint _value);
event Approval(address indexed _owner, address indexed _spender, uint _value);
```

The transfer function, in particular, will issue an event log containing the token sender, token receiver, and the value.

The canonical signature (hash) of

```
event Transfer(address indexed _from, address indexed _to, uint _value);
```

is exactly

```
0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef
```

This is the Transfer Topic Hash we saw in the previous slide.

Note that this hash is independent of tokens. It is the same for every token on Ethereum!

Ethereum Events

Sender and Receiver


```
{"address":"0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0","topics":
["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef","0x000000000000
000000000000d0a6e6c54dbc68db5db3a091b171a77407ff7ccf","0x000000000000000000000000ba80
9aeb7f910d44ece7f1f17af99b489d010ffc"],"data":"0x0000000000000000000000000000000000000000
0000000002143420340dabfb0de","blockNumber":"0x3c64e8","timeStamp":"0x5957a583","gasP
rice":"0x4e3b29200","gasUsed":"0x143ae","logIndex":"0x23","transactionHash":"0x081357
95a5c917b6eff494f5778f6b33e810fe6ed3646799a36d835aa3f94c64","transactionIndex":"0x1b"
}
```


There are two more “topics” in the Ethereum event log. They are simply the *from* and *to* addresses, respectfully.
In the above example,
0x000000000000000000000000d0a6e6c54dbc68db5db3a091b171a77407ff7ccf is the from address,
0x000000000000000000000000ba809aeb7f910d44ece7f1f17af99b489d010ffc is the to address.

Note that if you directly use these addresses, it will not work. Because the addresses have 0 paddings. You need to remove the zero paddings from the 3rd character to the 25th character.

What I would do in python is:

```
def hex_add_converter(add):
    '''
    Drop the 24 characters after 0x in the hex value of the topic addresses
    '''
    return add[0:2]+add[26:]
```

Ethereum Events

Value


```
{"address":"0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0","topics":
["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef","0x000000000000
0000000000d0a6e6c54dbc68db5db3a091b171a77407ff7ccf","0x0000000000000000000000ba80
9aeb7f910d44ece7f1f17af99b489d010ffc"],"data":"0x0000000000000000000000000000000000
00000000002143420340dabfb0de","blockNumber":"0x3c64e8","timeStamp":"0x5957a583","gasP
rice":"0x4e3b29200","gasUsed":"0x143ae","logIndex":"0x23","transactionHash":"0x081357
95a5c917b6eff494f5778f6b33e810fe6ed3646799a36d835aa3f94c64","transactionIndex":"0x1b"
}
```


The data field:

```
"data":"0x0000000000000000000000000000000000000000002143420340dabfb0de"
```

Indicates the value of the token transfer.

I have an interesting StackOverflow question on how to parse it.

The data field is a hex value. You need to use a converter to convert it to decimal. The above value, for example, should convert to 613588994158502064350.

On some ethereum websites, however, they see the values in 10^18. You can find another StackOverflow question on this matter as well.

Therefore, 613588994158502064350/10^18 = 613.588994159

Both notations should be acceptable.