

Trace methods

Infura provides access to the following trace API methods to allow users to gain insights into the execution of smart contracts and transactions:

- [trace_block](#)
- [trace_call](#)
- [trace_callMany](#)
- [trace_transaction](#)
- [trace_filter](#)

info Trace API is currently an open beta feature, available to paying Infura customers. When tracing transactions, the trace diagnostic options are [trace](#) and [stateDiff](#).

info `trace_block`, `trace_filter`, and `trace_transaction` return `trace` information, whereas `trace_call` and `trace_callMany` allow you to use the `trace` or `stateDiff` diagnostic options when tracing calls or transactions.

trace

Provides an ordered trace of the instructions executed by the Ethereum Virtual Machine (EVM) during the execution of a smart contract transaction. Excludes precompiled contracts.

Trace example `"trace": [{ "action": { "callType": "call", "from": "0xfe3b557e8fb62b89f4916b721be55ceb828dbd73", "gas": "0xffadea", "input": "0x", "to": "0x010000000000000000000000000000000000", "value": "0x0" }, "result": { "gasUsed": "0x1e", "output": "0x" }, "subtraces": 0, "traceAddress": [], "type": "call" }]` Key Value action Transaction details. `callType` Whether the transaction is `call` or `create`. `from` Address of the transaction sender. `gas` Gas provided by sender. `input` Transaction data. `to` Target of the transaction. `value` Value transferred in the transaction. `result` Transaction result. `gasUsed` Gas used by the transaction. Includes any refunds of unused gas. `output` Return value of the contract call. Contains only the actual value sent by a `RETURN` operation. If a `RETURN` was not executed, the output is empty bytes. `subTraces` Traces of contract calls made by the transaction. `traceAddress` Tree list address of where the call occurred, address of the parents, and order of the current sub call. `transactionHash` Hash of the transaction. `transactionPosition` Transaction position. `type` Whether the transaction is a `CALL` or `CREATE` series operation.

stateDiff

Displays state changes in the requested block for each transaction, represented as a map of accounts to an object. Lists the balance, code, nonce, and storage changes from immediately before the transaction to after the transaction. For the `key:value` pairs:

- `+`
 - indicates the field didn't exist before and now has the specified value.
- `-`
 - indicates a deleted value.
- `*`
 - has `from`
 - and `to`
 - value.

An absent value is distinct from zero when creating accounts or clearing storage. For example, when clearing storage, an absent value means that a particular storage slot has not yet been assigned a value, while a zero value means that the storage slot has been assigned and set to zero.

stateDiff example `"stateDiff": { "0xfe3b557e8fb62b89f4916b721be55ceb828dbd73": { "balance": { "": { "from": "0xfffffffffffffffffffffffff3e12a20b", "to": "0xfffffffffffffffffffffffff3dc5f091" } }, "code": "=", "nonce": { "": { "from": "0x14", "to": "0x15" } }, "storage": { } } } }` Key Value balance Change of balance event balance .from Balance before the transaction balance .to Balance after the transaction code Changes to code. None in this example nonce Change of nonce. nonce .from Nonce before the transaction. nonce .to Nonce after the transaction. storage Changes to storage. None in this example.

Last updated on Apr 19, 2024 [Previous eth_unsubscribe](#) [Next trace_block](#)