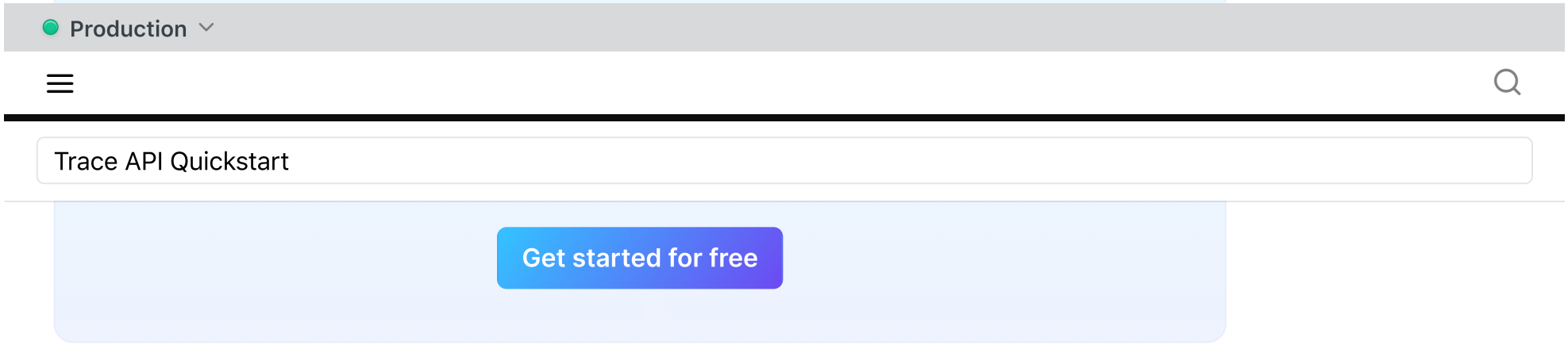
Trace API Quickstart

The Trace API provides insights into transaction processing and on-chain activity.

To use the Trace API your Alchemy plan must be set to the Growth or Enterprise tiers. Upgrade your plan for access. The Trace API is only supported on **Mainnet** and **Goerli**.



Introduction

The Trace API methods provide full externality trace functions on transactions executed in the Ethereum chain. Filtering is simple by using just the `address` information. The returned information includes the execution of the following:

- `CREATE`

- `SUICIDE`
- Variants of the `CALL`
- Input data
- Output data
- Gas usage
- Amount transferred
- Success status of individual actions

Trace use cases

Common use cases for the Trace API come in all flavors. Below are popular ones:

Use case	Description	Endpoint
Trace call	Returns the traces of the transaction.	`trace_call`
Trace block	Returns the traces executed in the block.	`trace_bock`
Trace get	Returns the traces executed in the block.	`trace_get`

Types of Traces

Type	Parameter	Description
Transaction Trace	`trace`	Trace of your transaction.
State difference	`stateDiff`	Ethereum state changed values of a transaction.
Virtual Machine Execution Trace	`vmTrace`	Runs a full trace of the VM's state & subcalls in the transaction.

Trace actions types

`CREATE`

- Used to create a smart contract.

Example response

JSON


```
{
    "action": {
      "from": "0x6090a6e47849629b7245dfa1ca21d94cd15878ef",
      "gas": "0x6a7f1",
      "init": "0x6060604052604051602080610516833981016040525 15b60028054600160a060020a038084166(
      "value": "0xe4b4b8af6a70000"
    },
    "blockHash": "0x6d00f7707938cca36b0730d8f7f090543242002b6fa0fe94bf85b9ab02e6bed6",
    "blockNumber": 4000036,
    "result": {
      "address": "0xfc9779d9a0f2715435a3e8ebf780322145d7546e",
      "code": "0x6060604052361561008576 3ffffffff60e060020a60003504166305b34410811461008a57806(
      "gasUsed": "0x52ce0"
    },
    "subtraces": 0,
    "traceAddress": [
      0
    ],
    "transactionHash": "0xc9601ea5ca42e57c3ef1d770ab0b278d6aadf2511a4feb879cba573854443423",
    "transactionPosition": 70,
    "type": "create"
  },
```


Create parameter definitions

Parameter	Description	Example

Parameter	Description	Example
`from`	The address that created the contract.	`0x6090a6e47849629b7245dfa1ca21d94cd15878ef`
`gas`	Contract gas fee to create the contract.	`0x6a7f1`
`init`	Initialization code to create the contract.	`0x60606040526040516020806105168339810160405...`
`value`	The value sent to the contract.	`0xe4b4b8af6a7000`
`blockHash`	Block hash ID in the transaction.	`0x6d00f7707938cca36b0730d8f7f090543242002b6fa0fe94bf85b9`
`blockNumber`	Transaction block number.	`4000036`
`address`	Location address of the new contract.	`0xfc9779d9a0f2715435a3e8ebf780322145d7546e`

Parameter	Description	Example
`code`	The contract code.	`0x6060604052361561008857 63ffffffff60e060020...`
`gasUsed`	Required fee (gas) to create the contract.	`0x52ce0`
`subtraces`	Total number of transaction child traces.	`0`
`traceAddress`	Index of a trace & order of subcalls.	`0`
`transactionHash`	Hash ID of the transaction.	`0xc9601ea5ca42e57c3ef1d770ab0b278d6aadf2511a4feb879cba5`
`transactionPosition`	The transaction position in the block.	`70`
`type`	Type of operation code (OPCODE).	`create`

`SUICIDE`

- `SUICIDE` is used by an owner of a smart contract to destroy the contract.
- `SUICIDE` transfers a contract's `address` balance clearing on-chain memory.
- The cleared memory is processed as a refund of the total gas cost to complete the transaction.

Example response

JSON


```
{
    "action": {
      "address": "0x87051f6ba0562fdb0485763562bf34cb2ad705b1",
      "refundAddress": "0x000000000000000000000000000000000000dead",
      "balance": "0x0"
    },
    "blockHash": "0x6d00f7707938cca36b0730d8f7f090543242002b6fa0fe94bf85b9ab02e6bed6",
    "blockNumber": 4000036,
    "result": null,
    "subtraces": 0,
    "traceAddress": [
      1,
      2,
      2
    ],
    "transactionHash": "0xbc15addb97490a168dc1d099ab8537caf2e4ff7d1deeff6d685d2d594a750037",
    "transactionPosition": 45,
    "type": "suicide"
  },
```


SUICIDE parameter definitions

Parameter	Description	Example

Parameter	Description	Example
`address`	The address of contract to destroy.	`0x87051f6ba0562fdb0485763562bf34cb2ad705b1`
`refundAddress`	Address to send remainder of contract `balance`.	`0x000000000000000000000000000000000000dead`
`blockHash`	Block hash ID of the transaction.	`0x6d00f7707938cca36b0730d8f7f090543242002b6fa0fe94bf8`
`blockNumber`	Transaction block number.	`4000036`
`result`	Used for `SELFDESTRUCT` calls.	`null`
`subtraces`	Total number of transaction child traces.	`0`
`traceAddress`	Index of the trace & the order of subcalls.	`1` and `2`
`transactionHash`	Hash ID of the transaction.	`0xbc15addb97490a168dc1d099ab8537caf2e4ff7d1deeff6d685`

Parameter	Description	Example
`transactionPosition`	The position of the transaction in the block.	`45`
`type`	Type of operation code (OPCODE).	`suicide`

`CALL`

- `CALL` is used for transferring ETH between externally owned accounts (EOAs).
- Also used to `CALL` a smart contract function.

Example response

JSON


```
{
    "action": {
      "from": "0xbc9f06dd67578b0b8b4d87fda9acde453bc4c067",
      "callType": "call",
      "gas": "0x97478",
      "input": "0xfebefd610000000000000000000000000000000000000000000000000000000000000040cc849
      "to": "0x6090a6e47849629b7245dfa1ca21d94cd15878ef",
      "value": "0x2386f26fc10000"
    },
    "blockHash": "0x6d00f7707938cca36b0730d8f7f090543242002b6fa0fe94bf85b9ab02e6bed6",
    "blockNumber": 4000036,
    "result": {
      "gasUsed": "0x7ad71",
      "output": "0x"
    },
    "subtraces": 4,
```

```
    "traceAddress": [],
    "transactionHash": "0x552b31a3a9c92577d65db62cf9f729e81571e10cad90e356423adcfa2caebacc",
    "transactionPosition": 71,
    "type": "call"
  }
```


`CALL` parameter definitions

Parameter	Description	Example
`from`	Address of the sender.	`0xbc9f06dd67578b0b8b4d87fda9acde453bc4c067`
`callType`	The type of `CALL`.	`call`, `delegatecall`, `callcode` or `staticcall`
`gas`	The gas included in the transaction.	`0x97478`
`input`	Specific call function on the contract.	`0xfebefd610000000000000000000000000000000000000000...`
`to`	Address of the receiver.	`0x6090a6e47849629b7245dfa1ca21d94cd15878ef`
`value`	Transferred value amount.	`0x2386f26fc10000`

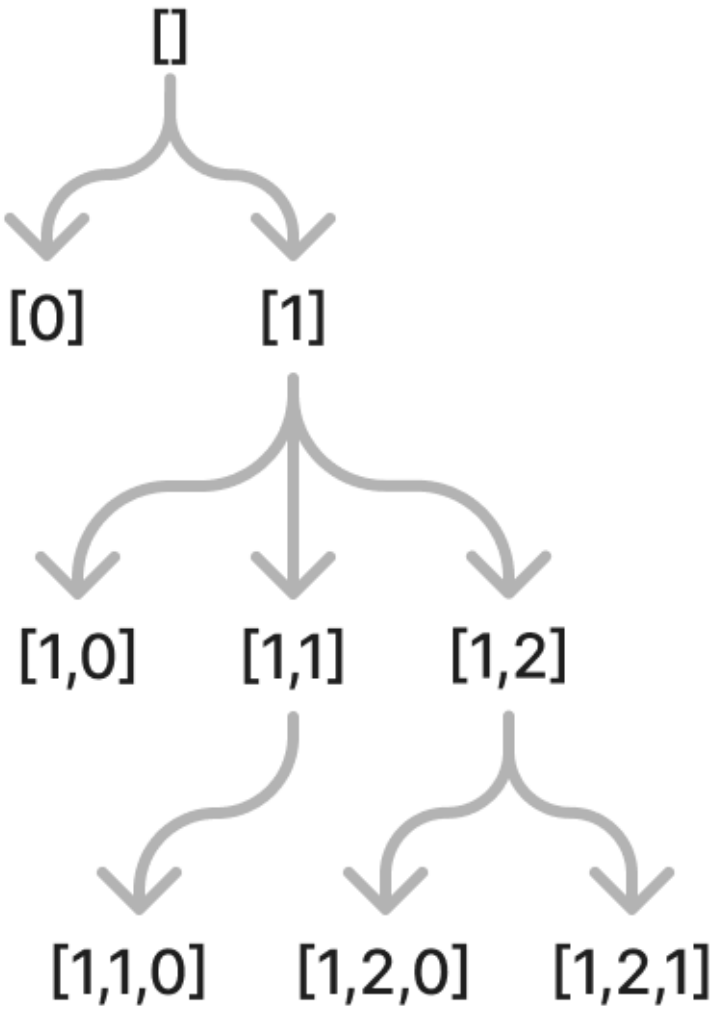
Parameter	Description	Example
`blockHash`	Transaction block hash ID.	`0x6d00f7707938cca36b0730d8f7f090543242002b6fa0fe94bf85b9`
`blockNumber`	Transaction block number.	`4000036`
`gasUsed`	gas used to execute the transaction.	`0x7ad71`
`output`	The result of the smart contract function call.	`0x`
`subtraces`	Total number of transaction child traces.	`4`
`traceAddress`	Index of a trace & the order of subcalls.	`[]`
`transactionHash`	Hash ID of the transaction.	`0x552b31a3a9c92577d65db62cf9f729e81571e10cad90e356423ad`

Parameter	Description	Example
`transactionPosition`	The position of the transaction in the block.	`71`
`type`	Type of operation code (OPCODE).	`call`

How to read a `traceAddress`

Traces are structured in a tree format. The `traceAddress` field represents the position of the trace in the tree. Below is a diagram of the `traceAddress` set of tree results:

Trace Tree



Trace tree diagram

Updated about 1 hour ago

← Delete Webhook

Trace API Endpoints →

Ethereum API
Polygon API
Arbitrum API
Optimism API
Solana API
NFT API
Transfers API
Token API
View all

Sign Up
Alchemy University
Login
Newsletter
Status
Goerli Faucet
Mumbai Faucet
Overviews
Gwei Calculator

COMPANY

About Us
Customers
Newsroom
Careers
Blog
Press Kit
Terms of Service

CONTACT

General Inquiries
Press
Sales
Discord
Email

alchemy
Certified Infrastructure