

Trace transactions

The `Trace API` is currently an open beta feature, available to paying Infura customers. Infura provides access to a trace API that allows you to get detailed transaction processing information. Use the API to extract information about contract interactions, transactions, and blocks on the Ethereum network. You can also use the API to retrieve transaction details that are not recorded on the blockchain. For example, use the [trace call](#) API to observe contract interactions.

Tip View the [trace API tutorial](#) for more information on how to call the API to trace, debug, analyze, and secure smart contracts. The trace API provides endpoints that can be categorized into the following groups, [ad-hoc tracing APIs](#) and [transaction-trace filtering APIs](#).

Ad-hoc tracing APIs

These API endpoints allow you to use the [trace](#) or [stateDiff](#) diagnostic options when tracing calls or transactions, and are helpful for debugging transactions and analyzing state changes.

info ThevmTrace diagnostic option is not supported. The ad-hoc tracing API endpoints are:

- [trace_call](#)
- [trace_callMany](#)

Transaction-trace filtering APIs

These API endpoints allow you to filter and search by specific information such as the block, address, or transaction. The endpoints only use the [trace](#) diagnostic option. The transaction-trace filtering API endpoints are:

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

Trace a transaction example

In the following example, you'll trace a transaction using a transaction hash on Ethereum mainnet. The result displays two traces, meaning two separate calls were made within the transaction, and includes the gas used for each call.

This example represents a call to the transfer function of an ERC-20 token contract (indicated in the input field), transferring tokens to the address, 0x7154980e9be95eab4eef2269650d4c7e17f156b8.

- cURL
- Result

[illegible]

Use cases

Use cases for the trace API include:

- Debugging transactions
 - The trace API allows you to analyze and debug Ethereum transactions. Trace the execution of a transaction to identify issues or bugs in smart contracts or dapps. Track the sequence of operations, inspect the input and output data, and pinpoint potential errors or unexpected behavior.
- Optimizing gas usage
 - The trace API allows you to analyze the gas consumption of transactions and identify areas where gas usage can be optimized. By examining the execution trace, you can spot expensive operations, inefficient code patterns, or unnecessary computations that consume excessive gas. You can use this information to refactor smart contracts and reduce transaction costs for users.
- Security auditing
 - Conduct security audits of smart contracts and dapps by tracing the execution flow. This allows you to identify potential vulnerabilities or attack vectors in the code. You can analyze contract interactions, track data modifications, and validate that the smart contract behaves as intended. This helps uncover security loopholes and ensures that the smart contracts are robust against various types of attacks, such as reentrancy or unauthorized access.

Last updated on Apr 19, 2024 [Previous](#) [Subscribe to events](#) [Next](#) [JSON-RPC methods](#)