

## Introduction

- [Alchemy API Reference Overview](#)
- - [Chain APIs Overview](#)
- - [Enhanced APIs Overview](#)
- [Alchemy Quickstart Guide](#)

## Resources

- [FAQ](#)
- - [Feature Support By Chain](#)
- - [Throughput](#)
- - [Batch Requests](#)
- - [Gas Limits](#)
- - [Error Reference](#)
- [Compute Units](#)
- - [Pricing Plans](#)
- - [Compute Unit Costs](#)

## NFT API

- [NFT API Quickstart](#)
- [NFT API Endpoints Overview](#)
- [NFT API FAQ](#)
- [Ownership & Token Gating](#)
- - [getNFTsForOwner.get](#)
- - [getOwnersForNFT.get](#)
- - [getOwnersForContract.get](#)
- - [isHolderOfContract.get](#)
- - [getContractsForOwner.get](#)
- - [getCollectionsForOwner.get](#)
- [NFT Metadata Access](#)
- - [getNFTMetadata.get](#)
- - [getNFTMetadataBatch.post](#)
- - [getContractMetadata.get](#)
- - [getCollectionMetadata.get](#)
- - [invalidateContract.get](#)
- - [getContractMetadataBatch.post](#)
- - [getNFTsForContract.get](#)
- - [getNFTsForCollection.get](#)
- - [searchContractMetadata.get](#)
- - [refreshNftMetadata.post](#)
- [Spam Detection](#)
- - [getSpamContracts.get](#)
- - [isSpamContract.get](#)
- - [isAirdropNFT.get](#)
- - [reportSpam.get](#)
- [Rarity Data](#)
- - [summarizeNFTAttributes.get](#)
- - [computeRarity.get](#)
- [Sales & Marketplace Data](#)
- - [getFloorPrice.get](#)
- - [getNFTSales.get](#)
- [NFT API V2 to V3 Migration Guide](#)
- - [NFT API V2 vs. V3 Endpoint Differences](#)
- [NFT API V2 Methods \(Older Version\)](#)
- - [getNFTs.get](#)
- - [getNFTMetadata.get](#)

- 
- [getNFTMetadataBatch.post](#)
- 
- [getContractMetadata.get](#)
- 
- [getContractMetadataBatch.post](#)
- 
- [getNFTsForCollection.get](#)
- 
- [getOwnersForToken.get](#)
- 
- [getOwnersForCollection.get](#)
- 
- [getSpamContracts.get](#)
- 
- [isSpamContract.get](#)
- 
- [isAirdrop.get](#)
- 
- [invalidateContract.get](#)
- 
- [getFloorPrice.get](#)
- 
- [computeRarity.get](#)
- 
- [searchContractMetadata.get](#)
- 
- [summarizeNFTAttributes.get](#)
- 
- [isHolderOfCollection.get](#)
- 
- [getNFTSales.get](#)
- 
- [getContractsForOwner.get](#)
- 
- [reportSpam.get](#)

## Transfers API (Tx History)

- [Transfers API Quickstart](#)
- [Transfers API Endpoints](#)
- 
- [alchemy\\_getAssetTransfers.post](#)

## Transaction Receipts API

- [Transaction Receipts Endpoints](#)
- 
- [alchemy\\_getTransactionReceipts.post](#)

## Token API

- [Token API Quickstart](#)
- [Token API Endpoints](#)
- 
- [alchemy\\_getTokenBalances.post](#)
- 
- [alchemy\\_getTokenMetadata.post](#)
- 
- [alchemy\\_getTokenAllowance.post](#)

## Subgraphs

- [Subgraphs Quickstart](#)
- [Supported Subgraph Chains](#)
- [Developing a Subgraph](#)
- 
- [Graph CLI](#)
- 
- [Creating a Subgraph](#)
- 
- [Project Structure](#)
- 
- [Data Sources](#)
- 
- [Writing Mappings](#)
- [Moving your Subgraph to Production](#)
- 
- [Deploying a Subgraph](#)
- 
- [Subgraph Versioning](#)
- 
- [Querying a Subgraph](#)
- 
- [Deleting a Subgraph](#)
- 
- [Direct Database Access](#)
- [Community subgraphs](#)

## Webhooks

- [Notify API Quickstart](#)
-

- [Notify Tutorials and Applications](#)
- 
- [Notify API FAQ](#)
- [Custom Webhooks Quickstart](#)
- 
- [Custom Webhooks FAQ](#)
- 
- [Custom Webhooks GraphQL Examples](#)
- 
- [Custom Webhook Filters](#)
- 
- [Custom Webhook Variables](#)
- [Custom Webhook API Methods](#)
- 
- [Read Variable Elements get](#)
- 
- [Create a Variable post](#)
- 
- [Delete a Variable delete](#)
- 
- [Update a Variable patch](#)
- [Notify API Methods](#)
- 
- [Get all webhooks get](#)
- 
- [Get all addresses for an Address Activity webhook get](#)
- 
- [Create webhook post](#)
- 
- [Add and remove webhook addresses patch](#)
- 
- [Replace webhook addresses put](#)
- 
- [Update webhook status put](#)
- 
- [Update webhook NFT filters patch](#)
- 
- [Update NFT metadata webhook filters patch](#)
- 
- [Get all webhook NFT filters get](#)
- 
- [Delete webhook delete](#)
- [Webhook Types](#)
- 
- [Custom Webhook](#)
- 
- [Address Activity Webhook](#)
- 
- [Mined Transaction Webhook](#)
- 
- [Dropped Transaction Webhook](#)
- 
- [NFT Activity Webhook](#)
- 
- [NFT Metadata Updates Webhook](#)

## Websockets

- [Subscription API Quickstart](#)
- [Best Practices for Using WebSockets in Web3](#)
- [Subscription API Endpoints](#)
- 
- [alchemy\\_minedTransactions](#)
- 
- [alchemy\\_pendingTransactions](#)
- 
- [newPendingTransactions](#)
- 
- [newHeads](#)
- 
- [logs](#)

## Trace API

- [Trace API Quickstart](#)
- [Trace API Endpoints](#)
- 
- [trace\\_block post](#)
- 
- [trace\\_call post](#)
- 
- [trace\\_get post](#)
- 
- [trace\\_rawTransaction post](#)
- 
- [trace\\_replayBlockTransactions post](#)
- 
- [trace\\_replayTransaction post](#)
- 
- [trace\\_transaction post](#)
- 
- [trace\\_filter post](#)
- [Trace API Resources](#)
-

- [What are EVM Traces?](#)
- 
- [Trace API vs. Debug API](#)
- 
- [What is trace\\_transaction?](#)
- 
- [What is trace\\_block?](#)
- 
- [What is trace\\_filter?](#)
- 
- [trace\\_call vs debug\\_traceCall](#)

## Debug API

- [Debug API Quickstart](#)
- [Debug API Endpoints](#)
- 
- [debug\\_traceCall post](#)
- 
- [debug\\_traceTransaction post](#)
- 
- [debug\\_traceBlockByNumber post](#)
- 
- [debug\\_traceBlockByHash post](#)

## ACCOUNT ABSTRACTION

- [Bundler API Quickstart](#)
- [Bundler API Endpoints](#)
- 
- [eth\\_getUserOperationReceipt post](#)
- 
- [eth\\_supportedEntryPoints post](#)
- 
- [eth\\_getUserOperationByHash post](#)
- 
- [eth\\_sendUserOperation post](#)
- 
- [rundler\\_maxPriorityFeePerGas post](#)
- 
- [eth\\_estimateUserOperationGas post](#)
- [Bundler API Fee Logic](#)
- [Factory Addresses](#)
- [Gas Manager Coverage API Quickstart](#)
- [Gas Manager Coverage API Endpoints](#)
- 
- [alchemy\\_requestPaymasterAndData post](#)
- 
- [alchemy\\_requestGasAndPaymasterAndData post](#)
- [Gas Manager Coverage API Fee Logic](#)
- [Gas Manager Deployment Addresses](#)
- [UserOperation Simulation Endpoints](#)
- 
- [alchemy\\_simulateUserOperationAssetChanges post](#)
- [AA-SDK](#)
- [Account Abstraction FAQ](#)

## Embedded Accounts

- [Accounts API Endpoints](#)
- 
- [Create Account post](#)
- 
- [Send Auth Email post](#)
- 
- [Authenticate User post](#)
- 
- [Get User post](#)
- 
- [Sign Message post](#)
- 
- [Register New Authenticator post](#)

## Gas Manager Admin API

- [Gas Manager Admin API Quickstart](#)
- [Gas Manager Admin API Endpoints](#)
- 
- [Create Policy post](#)
- 
- [Get Policy get](#)
- 
- [Delete Policy delete](#)
- 
- [Replace Policy put](#)
- 
- [Get All Policies get](#)
- 
- [Update Policy Status put](#)
- 
- [Get Policy Stats get](#)
- 
- [Get Sponsorships get](#)

## Alchemy Transact

- [Transact Quickstart](#)
- [Reinforced Transactions](#)
- [Transaction Simulation](#)
- - [Asset Changes](#)
- - [Execution Simulation](#)
- - [Bundle Simulation](#)
- - [Transaction Simulation Examples](#)
- - [Transaction Simulation FAQs](#)
- [Transaction Simulation Endpoints](#)
- - [alchemy\\_simulateAssetChanges.post](#)
- - [alchemy\\_simulateAssetChangesBundle.post](#)
- - [alchemy\\_simulateExecution.post](#)
- - [alchemy\\_simulateExecutionBundle.post](#)
- [Gas Optimized Transactions](#)
- - [alchemy\\_getGasOptimizedTransactionStatus.post](#)
- - [alchemy\\_sendGasOptimizedTransaction.post](#)
- [Private Transactions](#)
- - [eth\\_cancelPrivateTransaction.post](#)
- - [eth\\_sendPrivateTransaction.post](#)

## Alchemy SDK

- [Alchemy SDK Quickstart](#)
- - [How to use Alchemy SDK with Typescript](#)
  - [Examples Using the Alchemy SDK](#)
  - [How to Manage a Multichain Project Using Alchemy SDK](#)
- [Alchemy SDK Surface Overview](#)
- - [Alchemy SDK vs. Raw API Methods](#)
- [SDK Core Methods](#)
- - [call - SDK](#)
  - [send - SDK](#)
  - [estimateGas - SDK](#)
  - [findContractDeployer - SDK](#)
  - [getBalance - SDK](#)
  - [getBlock - SDK](#)
  - [getBlockNumber - SDK](#)
  - [getBlockWithTransactions - SDK](#)
  - [getCode - SDK](#)
  - [getFeeData - SDK](#)
  - [getGasPrice - SDK](#)
  - [getLogs - SDK](#)
  - [getStorageAt - SDK](#)
  - [getTokenBalances - SDK](#)
  - [getTokenMetadata - SDK](#)
  - [getTokensForOwner - SDK](#)
  - [getTransactionCount - SDK](#)
  - [getTransactionReceipt - SDK](#)
  - [getTransactionReceipts - SDK](#)
  - [isContractAddress - SDK](#)
  - [getAssetTransfers - SDK](#)
- [SDK NFT Methods](#)
- - [getNftsForOwner - SDK](#)
-

- [getNftMetadata - SDK](#)
- 
- [getNftMetadataBatch - SDK](#)
- 
- [refreshNftMetadata - SDK](#)
- 
- [getNftSales - SDK](#)
- 
- [searchContractMetadata - SDK](#)
- 
- [summarizeNftAttributes - SDK](#)
- 
- [getNftsForOwnerIterator - SDK](#)
- 
- [getNftsForContractIterator - SDK](#)
- 
- [getContractMetadata - SDK](#)
- 
- [getNftsForContract - SDK](#)
- 
- [getTransfersForOwner - SDK](#)
- 
- [getTransfersForContract - SDK](#)
- 
- [getMintedNfts - SDK](#)
- 
- [getOwnersForNft - SDK](#)
- 
- [getOwnersForContract - SDK](#)
- 
- [getSpamContracts - SDK](#)
- 
- [isSpamContract - SDK](#)
- 
- [refreshContract - SDK](#)
- 
- [getContractsForOwner - SDK](#)
- 
- [getFloorPrice - SDK](#)
- 
- [computeRarity - SDK](#)
- 
- [verifyNftOwnership - SDK](#)
- [SDK Transact Methods](#)
- 
- [getTransaction - SDK](#)
- 
- [sendTransaction - SDK](#)
- 
- [sendPrivateTransaction - SDK](#)
- 
- [cancelPrivateTransaction - SDK](#)
- 
- [waitForTransaction - SDK](#)
- 
- [estimateGas - SDK](#)
- 
- [getMaxPriorityFeePerGas - SDK](#)
- 
- [simulateAssetChanges - SDK](#)
- 
- [simulateAssetChangesBundle - SDK](#)
- 
- [simulateExecution - SDK](#)
- 
- [simulateExecutionBundle - SDK](#)
- [SDK Debug Methods](#)
- 
- [traceCall - SDK](#)
- 
- [traceTransaction - SDK](#)
- 
- [traceBlock - SDK](#)
- [SDK Notify Methods](#)
- 
- [getAllWebhooks - SDK](#)
- 
- [getAddresses - SDK](#)
- 
- [getNftFilters - SDK](#)
- 
- [createWebhook - SDK](#)
- 
- [updateWebhook - SDK](#)
- 
- [deleteWebhook - SDK](#)
- [SDK WebSockets Endpoints](#)
- [SDK Ethers Utils](#)
- 
- [arrayify](#)
- 
- [formatUnits](#)
- 
- [concat](#)
-

- [hexConcat](#)
- 
- [dnsEncode](#)
- 
- [hexDataLength](#)
- 
- [formatEther](#)
- 
- [hexDataSlice](#)
- 
- [hexStripZeros](#)
- 
- [hashMessage](#)
- 
- [isHexString](#)
- 
- [isValidName](#)
- 
- [joinSignature](#)
- 
- [splitSignature](#)
- 
- [toUtf8Bytes](#)
- 
- [hexValue](#)
- 
- [toUtf8String](#)
- 
- [hexZeroPad](#)
- 
- [zeroPad](#)
- 
- [hexlify](#)
- 
- [id](#)
- 
- [isBytes](#)
- 
- [isBytesLike](#)
- 
- [Interface](#)
- 
- [namehash](#)
- 
- [parseEther](#)
- 
- [parseUnits](#)
- 
- [stripZeros](#)
- 
- [Alchemy SDK V2 to V3 Migration Guide](#)
- 
- [Alchemy SDK V2 vs. V3 Method Differences](#)
- 
- [SDK V2 Methods](#)
- 
- [call - SDK](#)
- 
- [getAssetTransfers - SDK](#)
- 
- [getMintedNfts - SDK](#)
- 
- [verifyNftOwnership - SDK](#)
- 
- [getOwnersForNft - SDK](#)
- 
- [computeRarity - SDK](#)
- 
- [getTransfersForContract - SDK](#)
- 
- [getNftsForOwner - SDK](#)
- 
- [refreshContract - SDK](#)
- 
- [getOwnersForContract - SDK](#)
- 
- [getFloorPrice - SDK](#)
- 
- [isSpamContract - SDK](#)
- 
- [findContractDeployer - SDK](#)
- 
- [getSpamContracts - SDK](#)
- 
- [getGasPrice - SDK](#)
- 
- [getBalance - SDK](#)
- 
- [getBlock - SDK](#)
- 
- [getBlockWithTransactions - SDK](#)
- 
- [estimateGas - SDK](#)
- 
- [getBlockNumber - SDK](#)
- 
- [getCode - SDK](#)

- 
- [getFeeData - SDK](#)
- 
- [getLogs - SDK](#)
- 
- [getNftMetadataBatch - SDK](#)
- 
- [getTokensForOwner - SDK](#)
- 
- [getStorageAt - SDK](#)
- 
- [getTokenBalances - SDK](#)
- 
- [getTransactionCount - SDK](#)
- 
- [getTokenMetadata - SDK](#)
- 
- [getTransactionReceipt - SDK](#)
- 
- [send - SDK](#)
- 
- [getTransactionReceipts - SDK](#)
- 
- [getTransaction - SDK](#)
- 
- [isContractAddress - SDK](#)
- 
- [getNftMetadata - SDK](#)
- 
- [getNftSales - SDK](#)
- 
- [cancelPrivateTransaction - SDK](#)
- 
- [sendPrivateTransaction - SDK](#)
- 
- [traceTransaction - SDK](#)
- 
- [simulateExecutionBundle - SDK](#)
- 
- [simulateExecution - SDK](#)
- 
- [getMaxPriorityFeePerGas - SDK](#)
- 
- [simulateAssetChangesBundle - SDK](#)
- 
- [estimateGas - SDK](#)
- 
- [simulateAssetChanges - SDK](#)
- 
- [traceBlock - SDK](#)
- 
- [waitForTransaction - SDK](#)
- 
- [traceCall - SDK](#)
- 
- [sendTransaction - SDK](#)
- 
- [updateWebhook - SDK](#)
- 
- [refreshNftMetadata - SDK](#)
- 
- [createWebhook - SDK](#)
- 
- [getNftFilters - SDK](#)
- 
- [getAddresses - SDK](#)
- 
- [summarizeNftAttributes - SDK](#)
- 
- [deleteWebhook - SDK](#)
- 
- [searchContractMetadata - SDK](#)
- 
- [getAllWebhooks - SDK](#)
- 
- [getNftsForOwnerIterator - SDK](#)
- 
- [getNftsForContractIterator - SDK](#)
- 
- [getContractMetadata - SDK](#)
- 
- [getTransfersForOwner - SDK](#)
- 
- [getNftsForContract - SDK](#)

## Ethereum

- [Ethereum API Quickstart](#)
- [Ethereum API FAQ](#)
- 
- [Ethereum Developer Guide to the Merge](#)
- 
- [How to decode an eth\\_call response](#)
- 
- [How do I distinguish between a contract address and a wallet address?](#)



- [Ethereum API Endpoints](#)
- 
- [eth\\_blockNumber - Ethereum post](#)
- [eth\\_getBalance - Ethereum post](#)
- [eth\\_getLogs - Ethereum post](#)
- [eth\\_chainId - Ethereum post](#)
- [eth\\_getBlockByNumber - Ethereum post](#)
- [eth\\_accounts - Ethereum post](#)
- [eth\\_feeHistory - Ethereum post](#)
- [eth\\_estimateGas - Ethereum post](#)
- [eth\\_gasPrice - Ethereum post](#)
- [eth\\_getBlockTransactionCountByHash - Ethereum post](#)
- [eth\\_getBlockReceipts - Ethereum post](#)
- [eth\\_getBlockTransactionCountByNumber - Ethereum post](#)
- [eth\\_getCode - Ethereum post](#)
- [eth\\_getProof - Ethereum post](#)
- [eth\\_getStorageAt - Ethereum post](#)
- [eth\\_getTransactionByBlockHashAndIndex - Ethereum post](#)
- [eth\\_getTransactionByHash - Ethereum post](#)
- [eth\\_getTransactionCount - Ethereum post](#)
- [eth\\_getTransactionReceipt - Ethereum post](#)
- [eth\\_getUncleByBlockHashAndIndex - Ethereum post](#)
- [eth\\_getUncleByBlockNumberAndIndex - Ethereum post](#)
- [eth\\_getUncleCountByBlockHash - Ethereum post](#)
- [eth\\_getUncleCountByBlockNumber - Ethereum post](#)
- [eth\\_maxPriorityFeePerGas - Ethereum post](#)
- [eth\\_protocolVersion - Ethereum post](#)
- [eth\\_sendRawTransaction - Ethereum post](#)
- [net\\_listening - Ethereum post](#)
- [net\\_version - Ethereum post](#)
- [web3\\_clientVersion - Ethereum post](#)
- [web3\\_sha3 - Ethereum post](#)
- [eth\\_getTransactionByBlockNumberAndIndex - Ethereum post](#)
- [eth\\_call - Ethereum post](#)
- [eth\\_getBlockByHash - Ethereum post](#)
- [eth\\_createAccessList - Ethereum post](#)
- [eth\\_newFilter - Ethereum post](#)
- [eth\\_getFilterChanges - Ethereum post](#)
- [eth\\_getFilterLogs - Ethereum post](#)
- [eth\\_newBlockFilter - Ethereum post](#)
- [eth\\_newPendingTransactionFilter - Ethereum post](#)
- [eth\\_uninstallFilter - Ethereum post](#)
- [eth\\_subscribe](#)
- [eth\\_unsubscribe](#)

## Polygon PoS

- [Polygon PoS API Quickstart](#)
- 
- [Polygon SDK Examples](#)
- [Polygon PoS API FAQ](#)
- [Polygon PoS API Endpoints](#)
- 
- [bor\\_getAuthor - Polygon PoS post](#)

- 
- [bor\\_getCurrentProposer - Polygon PoS post](#)
- 
- [bor\\_getCurrentValidators - Polygon PoS post](#)
- 
- [bor\\_getRootHash - Polygon PoS post](#)
- 
- [eth\\_accounts - Polygon PoS post](#)
- 
- [eth\\_call - Polygon PoS post](#)
- 
- [eth\\_chainId - Polygon PoS post](#)
- 
- [eth\\_estimateGas - Polygon PoS post](#)
- 
- [eth\\_gasPrice - Polygon PoS post](#)
- 
- [eth\\_getBalance - Polygon PoS post](#)
- 
- [eth\\_getBlockByHash - Polygon PoS post](#)
- 
- [eth\\_getBlockByNumber - Polygon PoS post](#)
- 
- [eth\\_getBlockTransactionCountByHash - Polygon PoS post](#)
- 
- [eth\\_getBlockTransactionCountByNumber - Polygon PoS post](#)
- 
- [eth\\_getCode - Polygon PoS post](#)
- 
- [eth\\_getFilterChanges - Polygon PoS post](#)
- 
- [eth\\_getFilterLogs - Polygon PoS post](#)
- 
- [eth\\_getLogs - Polygon PoS post](#)
- 
- [eth\\_getRootHash - Polygon PoS post](#)
- 
- [eth\\_getSignersAtHash - Polygon PoS post](#)
- 
- [eth\\_getStorageAt - Polygon PoS post](#)
- 
- [eth\\_getTransactionByBlockHashAndIndex - Polygon PoS post](#)
- 
- [eth\\_getTransactionByBlockNumberAndIndex - Polygon PoS post](#)
- 
- [eth\\_getTransactionByHash - Polygon PoS post](#)
- 
- [eth\\_getTransactionCount - Polygon PoS post](#)
- 
- [eth\\_getTransactionReceipt - Polygon PoS post](#)
- 
- [eth\\_getTransactionReceiptsByBlock - Polygon PoS post](#)
- 
- [eth\\_sendRawTransaction - Polygon PoS post](#)
- 
- [eth\\_uninstallFilter - Polygon PoS post](#)
- 
- [net\\_listening - Polygon PoS post](#)
- 
- [eth\\_getUncleCountByBlockHash - Polygon PoS post](#)
- 
- [eth\\_getUncleCountByBlockNumber - Polygon PoS post](#)
- 
- [eth\\_newBlockFilter - Polygon PoS post](#)
- 
- [eth\\_newFilter - Polygon PoS post](#)
- 
- [eth\\_newPendingTransactionFilter - Polygon PoS post](#)
- 
- [web3\\_clientVersion - Polygon PoS post](#)
- 
- [eth\\_createAccessList - Polygon PoS post](#)
- 
- [eth\\_blockNumber - Polygon PoS post](#)
- 
- [bor\\_getSignersAtHash - Polygon PoS post](#)
- 
- [net\\_version - Polygon PoS post](#)
- 
- [eth\\_getProof - Polygon PoS post](#)
- 
- [eth\\_getUncleByBlockNumberAndIndex - Polygon PoS post](#)
- 
- [eth\\_subscribe - Polygon PoS](#)
- 
- [eth\\_unsubscribe - Polygon PoS](#)

## Polygon zkEVM

- [Polygon zkEVM API Quickstart](#)
- [Polygon zkEVM API FAQ](#)
- 
- [What is the difference between Polygon zkEVM and Ethereum?](#)
- 
- [What is the difference between Polygon zkEVM and Polygon PoS?](#)

- [Polygon zkEVM Endpoints](#)
- 
- [eth\\_getTransactionCount](#) - Polygon zkEVM post
- 
- [eth\\_call](#) - Polygon zkEVM post
- 
- [eth\\_chainId](#) - Polygon zkEVM post
- 
- [eth\\_newBlockFilter](#) - Polygon zkEVM post
- 
- [eth\\_estimateGas](#) - Polygon zkEVM post
- 
- [eth\\_newFilter](#) - Polygon zkEVM post
- 
- [eth\\_gasPrice](#) - Polygon zkEVM post
- 
- [eth\\_sendRawTransaction](#) - Polygon zkEVM post
- 
- [eth\\_getBalance](#) - Polygon zkEVM post
- 
- [eth\\_uninstallFilter](#) - Polygon zkEVM post
- 
- [eth\\_getBlockByHash](#) - Polygon zkEVM post
- 
- [net\\_version](#) - Polygon zkEVM post
- 
- [eth\\_getBlockByNumber](#) - Polygon zkEVM post
- 
- [web3\\_clientVersion](#) - Polygon zkEVM post
- 
- [eth\\_getBlockTransactionCountByHash](#) - Polygon zkEVM post
- 
- [eth\\_getBlockTransactionCountByNumber](#) - Polygon zkEVM post
- 
- [zkevm\\_batchNumber](#) - Polygon zkEVM post
- 
- [eth\\_getCode](#) - Polygon zkEVM post
- 
- [eth\\_getFilterChanges](#) - Polygon zkEVM post
- 
- [eth\\_getFilterLogs](#) - Polygon zkEVM post
- 
- [zkevm\\_getBatchByNumber](#) - Polygon zkEVM post
- 
- [eth\\_getLogs](#) - Polygon zkEVM post
- 
- [zkevm\\_getBroadcastURI](#) - Polygon zkEVM post
- 
- [eth\\_getStorageAt](#) - Polygon zkEVM post
- 
- [zkevm\\_isBlockConsolidated](#) - Polygon zkEVM post
- 
- [eth\\_getTransactionByBlockHashAndIndex](#) - Polygon zkEVM post
- 
- [zkevm\\_isBlockVirtualized](#) - Polygon zkEVM post
- 
- [eth\\_getTransactionByBlockNumberAndIndex](#) - Polygon zkEVM post
- 
- [zkevm\\_verifiedBatchNumber](#) - Polygon zkEVM post
- 
- [eth\\_getTransactionByHash](#) - Polygon zkEVM post
- 
- [zkevm\\_virtualBatchNumber](#) - Polygon zkEVM post
- 
- [eth\\_getCompilers](#) - Polygon zkEVM post
- 
- [eth\\_getUncleByBlockHashAndIndex](#) - Polygon zkEVM post
- 
- [eth\\_getUncleByBlockNumberAndIndex](#) - Polygon zkEVM post
- 
- [eth\\_getUncleCountByBlockHash](#) - Polygon zkEVM post
- 
- [eth\\_getUncleCountByBlockNumber](#) - Polygon zkEVM post
- 
- [eth\\_protocolVersion](#) - Polygon zkEVM post
- 
- [eth\\_blockNumber](#) - Polygon zkEVM post
- 
- [eth\\_getTransactionReceipt](#) - Polygon zkEVM post
- 
- [zkevm\\_batchNumberByBlockNumber](#) - Polygon zkEVM post
- 
- [zkevm\\_consolidatedBlockNumber](#) - Polygon zkEVM post
- 
- [zkevm\\_estimateFee](#) API - Polygon zkEVM post
- 
- [zkevm\\_estimateGasPrice](#) API - Polygon zkEVM post

## Arbitrum

- [Arbitrum API Quickstart](#)
- 
- [Arbitrum SDK Examples](#)
- [Arbitrum API FAQ](#)
-

- [Arbitrum vs. Ethereum API Differences](#)
- [Arbitrum API Endpoints](#)
- 
- [eth\\_call - Arbitrum post](#)
- 
- [eth\\_estimateGas - Arbitrum post](#)
- 
- [eth\\_accounts - Arbitrum post](#)
- 
- [eth\\_blockNumber - Arbitrum post](#)
- 
- [eth\\_chainId - Arbitrum post](#)
- 
- [eth\\_gasPrice - Arbitrum post](#)
- 
- [eth\\_getBalance - Arbitrum post](#)
- 
- [eth\\_getBlockTransactionCountByHash - Arbitrum post](#)
- 
- [eth\\_getBlockTransactionCountByNumber - Arbitrum post](#)
- 
- [eth\\_getCode - Arbitrum post](#)
- 
- [eth\\_getFilterChanges - Arbitrum post](#)
- 
- [eth\\_getFilterLogs - Arbitrum post](#)
- 
- [eth\\_getLogs - Arbitrum post](#)
- 
- [eth\\_getStorageAt - Arbitrum post](#)
- 
- [eth\\_getTransactionByBlockHashAndIndex - Arbitrum post](#)
- 
- [eth\\_getTransactionCount - Arbitrum post](#)
- 
- [eth\\_getUncleByBlockNumberAndIndex - Arbitrum post](#)
- 
- [eth\\_getUncleCountByBlockHash - Arbitrum post](#)
- 
- [eth\\_getUncleCountByBlockNumber - Arbitrum post](#)
- 
- [eth\\_newBlockFilter - Arbitrum post](#)
- 
- [eth\\_newFilter - Arbitrum post](#)
- 
- [eth\\_newPendingTransactionFilter - Arbitrum post](#)
- 
- [eth\\_uninstallFilter - Arbitrum post](#)
- 
- [net\\_listening - Arbitrum post](#)
- 
- [net\\_version - Arbitrum post](#)
- 
- [web3\\_clientVersion - Arbitrum post](#)
- 
- [web3\\_sha3 - Arbitrum post](#)
- 
- [eth\\_sendRawTransaction - Arbitrum post](#)
- 
- [eth\\_createAccessList - Arbitrum post](#)
- 
- [eth\\_maxPriorityFeePerGas - Arbitrum post](#)
- 
- [eth\\_feeHistory - Arbitrum post](#)
- 
- [eth\\_getBlockByHash - Arbitrum post](#)
- 
- [eth\\_getBlockByNumber - Arbitrum post](#)
- 
- [eth\\_getTransactionByBlockNumberAndIndex - Arbitrum post](#)
- 
- [eth\\_getTransactionByHash - Arbitrum post](#)
- 
- [eth\\_getProof - Arbitrum post](#)
- 
- [eth\\_getTransactionReceipt - Arbitrum post](#)
- 
- [eth\\_getUncleByBlockHashAndIndex - Arbitrum post](#)
- 
- [eth\\_subscribe](#)
- 
- [eth\\_unsubscribe](#)

## Optimism

- [Optimism API Quickstart](#)
- 
- [Optimism SDK Examples](#)
- [Optimism API FAQ](#)
- 
- [Optimism Error Codes](#)
- [Optimism API Endpoints](#)
- 
- [eth\\_call - Optimism post](#)
-

- [eth\\_estimateGas - Optimism post](#)
- 
- [eth\\_accounts - Optimism post](#)
- 
- [eth\\_blockNumber - Optimism post](#)
- 
- [eth\\_chainId - Optimism post](#)
- 
- [eth\\_gasPrice - Optimism post](#)
- 
- [eth\\_getBalance - Optimism post](#)
- 
- [eth\\_getBlockTransactionCountByHash - Optimism post](#)
- 
- [eth\\_getBlockTransactionCountByNumber - Optimism post](#)
- 
- [eth\\_getCode - Optimism post](#)
- 
- [eth\\_getFilterChanges - Optimism post](#)
- 
- [eth\\_getFilterLogs - Optimism post](#)
- 
- [eth\\_getLogs - Optimism post](#)
- 
- [eth\\_getStorageAt - Optimism post](#)
- 
- [eth\\_getTransactionByBlockHashAndIndex - Optimism post](#)
- 
- [eth\\_getTransactionByBlockNumberAndIndex - Optimism post](#)
- 
- [eth\\_getTransactionByHash - Optimism post](#)
- 
- [eth\\_getTransactionCount - Optimism post](#)
- 
- [eth\\_getTransactionReceipt - Optimism post](#)
- 
- [eth\\_getUncleByBlockHashAndIndex - Optimism post](#)
- 
- [eth\\_getUncleByBlockNumberAndIndex - Optimism post](#)
- 
- [eth\\_getUncleCountByBlockHash - Optimism post](#)
- 
- [eth\\_getUncleCountByBlockNumber - Optimism post](#)
- 
- [eth\\_newBlockFilter - Optimism post](#)
- 
- [eth\\_newFilter - Optimism post](#)
- 
- [eth\\_newPendingTransactionFilter - Optimism post](#)
- 
- [eth\\_protocolVersion - Optimism post](#)
- 
- [eth\\_sendRawTransaction - Optimism post](#)
- 
- [eth\\_syncing - Optimism post](#)
- 
- [eth\\_uninstallFilter - Optimism post](#)
- 
- [net\\_listening - Optimism post](#)
- 
- [net\\_version - Optimism post](#)
- 
- [web3\\_clientVersion - Optimism post](#)
- 
- [web3\\_sha3 - Optimism post](#)
- 
- [eth\\_getBlockByHash - Optimism post](#)
- 
- [eth\\_getBlockByNumber - Optimism post](#)
- 
- [eth\\_getProof - Optimism post](#)
- 
- [eth\\_subscribe](#)
- 
- [eth\\_unsubscribe](#)

## Base

- [Base API Quickstart](#)
- [Base API FAQ](#)
- [Base API Endpoints](#)
- 
- [eth\\_accounts - Base post](#)
- 
- [eth\\_blockNumber - Base post](#)
- 
- [eth\\_call - Base post](#)
- 
- [eth\\_chainId - Base post](#)
- 
- [eth\\_estimateGas - Base post](#)
- 
- [eth\\_feeHistory - Base post](#)
- 
- [eth\\_gasPrice - Base post](#)

- 
- [eth\\_getBalance - Base post](#)
- 
- [eth\\_getBlockByHash - Base post](#)
- 
- [eth\\_getBlockByNumber - Base post](#)
- 
- [eth\\_getBlockTransactionCountByHash - Base post](#)
- 
- [eth\\_getBlockTransactionCountByNumber - Base post](#)
- 
- [eth\\_getCode - Base post](#)
- 
- [eth\\_getFilterChanges - Base post](#)
- 
- [eth\\_getFilterLogs - Base post](#)
- 
- [eth\\_getLogs - Base post](#)
- 
- [eth\\_getProof - Base post](#)
- 
- [eth\\_getStorageAt - Base post](#)
- 
- [eth\\_getTransactionByBlockHashAndIndex - Base post](#)
- 
- [eth\\_getTransactionByBlockNumberAndIndex - Base post](#)
- 
- [eth\\_getTransactionByHash - Base post](#)
- 
- [eth\\_getTransactionCount - Base post](#)
- 
- [eth\\_getTransactionReceipt - Base post](#)
- 
- [eth\\_getUncleByBlockHashAndIndex - Base post](#)
- 
- [eth\\_getUncleByBlockNumberAndIndex - Base post](#)
- 
- [eth\\_getUncleCountByBlockHash - Base post](#)
- 
- [eth\\_getUncleCountByBlockNumber - Base post](#)
- 
- [eth\\_maxPriorityFeePerGas - Base post](#)
- 
- [eth\\_newBlockFilter - Base post](#)
- 
- [eth\\_newFilter - Base post](#)
- 
- [eth\\_newPendingTransactionFilter - Base post](#)
- 
- [eth\\_protocolVersion - Base post](#)
- 
- [eth\\_sendRawTransaction - Base post](#)
- 
- [eth\\_syncing - Base post](#)
- 
- [eth\\_uninstallFilter - Base post](#)
- 
- [net\\_listening - Base post](#)
- 
- [web3\\_sha3 - Base post](#)

## \* Solana

- [Solana API Quickstart](#)
- [Solana API FAQ](#)
- [Solana API Endpoints](#)
- 
- [getAccountInfo post](#)
- 
- [simulateTransaction post](#)
- 
- [getBalance post](#)
- 
- [getBlock post](#)
- 
- [getBlockCommitment post](#)
- 
- [getBlockProduction post](#)
- 
- [getBlocks post](#)
- 
- [getBlocksWithLimit post](#)
- 
- [getBlockTime post](#)
- 
- [getClusterNodes post](#)
- 
- [getEpochInfo post](#)
- 
- [getEpochSchedule post](#)
- 
- [getFeeForMessage post](#)
- 
- [getFirstAvailableBlock post](#)
-

- [getGenesisHash post](#)
- 
- [getHealth post](#)
- 
- [getHighestSnapshotSlot post](#)
- 
- [getIdentity post](#)
- 
- [getInflationGovernor post](#)
- 
- [getInflationRate post](#)
- 
- [getInflationReward post](#)
- 
- [getLargestAccounts post](#)
- 
- [getMaxRetransmitSlot post](#)
- 
- [getMaxShredInsertSlot post](#)
- 
- [getMinimumBalanceForRentExemption post](#)
- 
- [getMultipleAccounts post](#)
- 
- [getProgramAccounts post](#)
- 
- [getRecentPerformanceSamples post](#)
- 
- [getSignaturesForAddress post](#)
- 
- [getSignatureStatuses post](#)
- 
- [getSlot post](#)
- 
- [getSlotLeader post](#)
- 
- [getSlotLeaders post](#)
- 
- [getSupply post](#)
- 
- [getTokenAccountBalance post](#)
- 
- [getTokenAccountsByOwner post](#)
- 
- [getTokenSupply post](#)
- 
- [getTransaction post](#)
- 
- [getVersion post](#)
- 
- [getVoteAccounts post](#)
- 
- [isBlockhashValid post](#)
- 
- [minimumLedgerSlot post](#)
- 
- [sendTransaction post](#)
- 
- [requestAirdrop post](#)
- 
- [getBlockHeight post](#)
- 
- [getRecentBlockhash post](#)

## Astar

- [Astar API Quickstart](#)
- [Astar API FAQ](#)
- [Astar API Endpoints](#)
- 
- [eth\\_accounts - Astar post](#)
- 
- [eth\\_getTransactionReceipt - Astar post](#)
- 
- [eth\\_maxPriorityFeePerGas - Astar post](#)
- 
- [eth\\_blockNumber - Astar post](#)
- 
- [eth\\_call - Astar post](#)
- 
- [eth\\_chainId - Astar post](#)
- 
- [eth\\_gasPrice - Astar post](#)
- 
- [eth\\_getBalance - Astar post](#)
- 
- [eth\\_getBlockByHash - Astar post](#)
- 
- [eth\\_getBlockByNumber - Astar post](#)
- 
- [eth\\_getBlockTransactionCountByHash - Astar post](#)
- 
- [eth\\_getBlockTransactionCountByNumber - Astar post](#)
- 
- [eth\\_getCode - Astar post](#)

- 
- [eth\\_getStorageAt - Astar post](#)
- 
- [eth\\_getTransactionByBlockHashAndIndex - Astar post](#)
- 
- [eth\\_getTransactionByBlockNumberAndIndex - Astar post](#)
- 
- [eth\\_getTransactionByHash - Astar post](#)
- 
- [eth\\_getTransactionCount - Astar post](#)
- 
- [eth\\_getUncleByBlockNumberAndIndex - Astar post](#)
- 
- [eth\\_sendRawTransaction - Astar post](#)
- 
- [net\\_version - Astar post](#)
- 
- [web3\\_clientVersion - Astar post](#)
- 
- [web3\\_sha3 - Astar post](#)
- 
- [eth\\_getLogs - Astar post](#)
- 
- [eth\\_getFilterChanges - Astar post](#)
- 
- [eth\\_getFilterLogs - Astar post](#)
- 
- [eth\\_newFilter - Astar post](#)
- 
- [eth\\_newPendingTransactionFilter - Astar post](#)
- 
- [eth\\_uninstallFilter - Astar post](#)
- 
- [eth\\_newBlockFilter - Astar post](#)
- 
- [eth\\_estimateGas - Astar post](#)
- 
- [eth\\_subscribe](#)
- 
- [eth\\_unsubscribe](#)

## STARKNET

- [Starknet API Quickstart](#)
- [Starknet API FAQ](#)
- [Starknet API Endpoints](#)
- 
- [starknet\\_addDeclareTransaction post](#)
- 
- [starknet\\_getClassAt post](#)
- 
- [starknet\\_addDeployAccountTransaction post](#)
- 
- [starknet\\_getClassHashAt post](#)
- 
- [starknet\\_addInvokeTransaction post](#)
- 
- [starknet\\_getEvents post](#)
- 
- [starknet\\_blockHashAndNumber post](#)
- 
- [starknet\\_getNonce post](#)
- 
- [starknet\\_blockNumber post](#)
- 
- [starknet\\_getStateUpdate post](#)
- 
- [starknet\\_call post](#)
- 
- [starknet\\_getStorageAt post](#)
- 
- [starknet\\_chainId post](#)
- 
- [starknet\\_getTransactionByBlockIdAndIndex post](#)
- 
- [starknet\\_estimateFee post](#)
- 
- [starknet\\_getTransactionByHash post](#)
- 
- [starknet\\_getBlockTransactionCount post](#)
- 
- [starknet\\_getTransactionReceipt post](#)
- 
- [starknet\\_getBlockWithTxHashes post](#)
- 
- [starknet\\_pendingTransactions post](#)
- 
- [starknet\\_getBlockWithTxs post](#)
- 
- [starknet\\_syncing post](#)
- 
- [starknet\\_getClass post](#)
- 
- [starknet\\_estimateMessageFee post](#)



# trace\_call vs debug\_traceCall

The differences between the trace\_call method by OpenEthereum and the debug\_traceCall method by Geth

## Introduction

Geth and OpenEthereum are two popular Ethereum clients. In this article, we'll compare and contrast the [trace\\_call](#) method offered by OpenEthereum with the [debug\\_traceCall](#) method offered by Geth. Both of these methods are used for transaction tracing.

### NOTE

The trace\_call method was initially supported by the OpenEthereum client but now OpenEthereum has been deprecated and a new client Erigon is supporting the trace methods.

Going forward, we will only use the name of the Erigon Ethereum client when associating with the trace methods.

## Prerequisites

Before reading this article you should know about [Ethereum Clients](#) and [EVM Traces](#) .

## The trace\_call method

The [trace\\_call](#) method executes the given call (transaction) and returns a number of possible traces for it. It's helpful for debugging transactions and analyzing state changes due to a transaction. Under the hood, [trace\\_call](#) is only supported by OpenEthereum or Erigon clients, but if you're using an Alchemy API key we'll automatically route the request for you so you don't have to worry about the node client.

Here are the parameters and response payloads for trace\_call

### Parameters

1. Object
2.
  - Call options.
3.
  - from
4.
  - :
5.
  - Address
6.
  - - (optional) - The address the transaction is sent from.
7.
  - to
8.
  - :
9.
  - Address
10.
  - - (optional when creating a new contract) - The address the transaction is directed to.
11.
  - gas
12.
  - :
13.
  - Quantity
14.
  - - (optional) Integer formatted as a hex string that represents the gas provided for the transaction execution.
15.
  - gasPrice
16.
  - :
17.
  - Quantity
18.
  - - (optional) Integer formatted as a hex string that represents the gas price used for each paid gas.
19.
  - value
20.
  - :
21.
  - Quantity
22.
  - - (optional) Integer formatted as a hex string that represents the value sent with the given transaction.
23.
  - data
24.
  - :
25.
  - Data
26.
  - - (optional) 4-byte hash of the method signature followed by encoded parameters. This basically represents which function to call in the smart contract along with the function parameters, if it's just a value transfer then this field can be empty.

- vmTrace is no longer supported by Alchemy. 1. Quantity 2. or 3. Tag 4. - (optional) Integer formatted as a hex string that represents a block number, or the string 5. 'earliest' 6. or 7. 'latest' 8. . If this parameter is supplied, the transaction is replayed in the context of the given parameter. 1. latest 9. 2. : The latest block that the client has observed. 10. 3. earliest 11. 4. : The lowest number block that the client has available. Intuitively, you can think of this as the first block created.

The `trace_call` method returns an array of traces. The structure of these traces is explained in [Types of Trace Actions and their Structure](#). Below you can find an example.

```
CURL ethers web3py curl https://eth-mainnet.g.alchemy.com/v2/demo -X POST -H "Content-Type: application/json" -d '{"method": "trace_call", "params": [{"from": "0x6f1FB6EFDf50F34bFA3F2bC0E557EdD71631638", "to": "0x6b175474e89094c44da98b954eedeac495271d0f", "value": "0x0", "data": "0x70a0823100000000000000000000000000000000000000000000000000000000E0d01A76C3Cf4288372a29124A26D4353EE51BE"}], ["trace"]}, {"id": 1, "jsonrpc": "2.0"}] const ethers = require ("ethers");

(async () => {const provider = new ethers.providers.JsonRpcProvider("https://eth-mainnet.g.alchemy.com/v2/demo");const response = await provider.send ("trace_call", [ {from: "0x6f1FB6EFDf50F34bFA3F2bC0E557EdD71631638" , "to": "0x6b175474e89094c44da98b954eedeac495271d0f", "data": "0x70a0823100000000000000000000000000000000000000000000000000000000E0d01A76C3Cf4288372a29124A26D4353EE51BE" } , ["trace"], );console.log (response);})(); from web3 import HTTPProvider client = HTTPProvider ('https://eth-mainnet.g.alchemy.com/v2/demo') result = client.make_request('trace_call' , [{"to": "0x6b175474e89094c44da98b954eedeac495271d0f", "data": "0x70a0823100000000000000000000000000000000000000000000000000000000E0d01A76C3Cf4288372a29124A26D4353EE51BE"}, ["trace", "latest"]})print (result)
```

[illegible]

The `debug_traceCall` method executes the given call (transaction) and returns a number of possible traces for it. It's helpful for debugging transactions and analyzing state changes due to a transaction. Under the hood, `debug_traceCall` is only supported by OpenEthereum or Erigon clients, but if you're using an Alchemy API key we'll automatically route the request for you so you don't have to worry about the node client.

Here are the parameters and response payloads for `debug_traceCall`

The parameters for the `debug_traceCall` method are:

1. Object
2.
  - Transaction object with the following fields: \* from
3.
  - - string - The address the transaction is sent from.
4.
  - to
5.
  - - string,[required]
6.
  - - The address the transaction is directed to.
7.
  - gasPrice
8.
  - - string - Integer of the
9.
  - gasPrice
10.
  - used for each paid gas.
11.
  - value
12.
  - - string - Integer of the value sent with the given transaction.

13.
  - data
14.
  - - string - Hash of the method signature and encoded parameters.
15. String
16.
  - One of the following options:1. block hash
17.
  1. block number
18.
  1. (in hex)
19.
  1. block tag
20.
  1. (one of the following):\* pending
21.
  1.
    - - A sample next block built by the client on top of the latest and containing the set of transactions usually taken from the local mempool. Intuitively, you can think of these as blocks that have not been mined yet.
22.
  1.
    - latest
23.
  1.
    - - The most recent block observed by the client, this block may be re-orged out of the canonical chain even under healthy/normal conditions.
24.
  1.
    - safe
25.
  1.
    - - The most recent crypto-economically secure block, cannot be re-orged outside of manual intervention driven by community coordination. Intuitively, this block is “unlikely” to be re-orged.Only available on Ethereum Goerli
26.
  1.
    - .
27.
  1.
    - finalized
28.
  1.
    - - The most recent crypto-economically secure block, that has been accepted by >2/3 of validators. Cannot be re-orged outside of manual intervention driven by community coordination. Intuitively, this block is very unlikely to be re-orged.Only available on Ethereum Goerli
29.
  1.
    - .
30.
  1.
    - earliest
31.
  1.
    - - The lowest numbered block the client has available. Intuitively, you can think of this as the first block created.
32. Object - tracer
33. tracer
34. - String to specify the type of tracer. Currently supports
35. callTracer
36. and
37. prestateTracer
38. (see below for definitions).
39. tracerConfig
40.
  - Object to specify configurations for the tracer\* onlyTopCall
41.
  - - boolean - setting this to
42.
  - true
43.
  - will only trace the main (top-level) call and none of the sub-calls. This avoids extra processing for each call frame if only the top-level call info is required (useful for getting
44.
  - revertReason
45.
  - ).

## callTracer

The callTracer tracks all the call frames executed during a transaction. The result will be a nested list of call frames. They form a tree with the top-level call at the root and sub-calls as children of the higher levels.

It's similar to the trace option of the [trace\\_call](#) method. Each call frame has the following fields:

field type description type string CALL, CREATE or SUICIDE from string address to string address value string hex-encoded amount of value transfer gas string hex-encoded gas provided for call gasUsed string hex-encoded gas used during call input string call data output string return data error string error, if any revertReason string Solidity revert reason, if any calls array of call frames list of sub-calls

## prestateTracer

The prestateTracer replays the transaction and tracks every part of state that is touched during that transaction.

field type description balance string balance in wei nonce uint64 nonce code string hex-encoded bytecode storage map[string]string storage slots of the contract As you've seen that the `callTracer` and `prestateTracer` options of `debug_traceCall` are similar to `trace` and `stateDiff` options of `debug_traceCall`, here's a table depicting this:

### Example Request and Response

```
URL ethers.web3py curl https://eth-mainnet.g.alchemy.com/v2/demo -X POST -H "Content-Type: application/json" --data '{"method": "debug_traceCall", "params": [{"from": null, "to": "0xb617547e89094c44da98b954eedeac495271d0f", "data": "0x70a08231000000000000000000000000E0d01A76C3Cf4288372a29124A26D4353EE51BE"}, {"latest"}], "id": "1", "jsonrpc": "2.0"] const ethers = require("ethers"); { async () => [const provider = new ethers.providers.JsonRpcProvider('https://eth- mainnet.g.alchemy.com/v2/demo'); console response = await provider.send ("debug_traceCall", { from: "0xe5cb067e90d5cd1f8052bf383562ae670ba4a211a8", : "0xcd66567ea99b07fa10730459537620B57ce903287", data: "0xae169a50000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 ", }, "0xF18CE", {tracer: "callTracer", tracerConfig: {onlyTopCall: false }}, )]; console.log(response); })(); from web3 import Web3, HTTPProvider provider = Web3.HTTPProvider ('https://eth-mainnet.g.alchemy.com/v2/demo') result=provider.make_request('debug_traceCall', [{to:"0xb617547e89094c44da98b954eedeac495271d0f", "data": "0x70a08231000000000000000000000000000000E0d01A76C3Cf4288372a29124A26D4353EE51BE"}, {"latest"}])print(result)
```

[illegible]

## Difference between

Since [trace\\_call](#) is OpenEthereum's (Now Erigon's) equivalent to Geth's [debug\\_traceCall](#) there are not many differences between them but there are some minor differences which are mentioned below:

- ```
ethers // Request using ethers.js in node.js const ethers = require ("ethers");
```

[illegible][illegible]

## Conclusion

Updated 5 months ago

What is trace filter? Debug API Quickstart Did this page help you? Yes No

