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

Custom Webhook Filters

Understand what filters are available for Custom Webhooks and how to use them

Alchemy's Custom Webhooks allow you to access any on-chain activity while also precisely defining filters to control your data pipelines. With Alchemy's filters, you are able to set up logic for when you want blockchain data, whether that's for token movement, transactions of interest, or full blocks of data! Additionally, they allow you to easily filter down the data to exactly what you need, saving you on data ingestion costs and reducing infrastructure overhead.

To get started with the filters head over to theCustom Webhook playground and test them out!

Alchemy's Custom Webhooks currently feature filters on 3 types of on-chain data:

- Events/Logs
- External Transactions
- Internal Transactions (Debug trace calls)

Event/Log Filters

Custom Webhook log/event filters leverage the same semantics as a traditionaleth_getLogs RPC call. In particular, we maintain the 2 key fields thateth_getLogs accepts to query for events of interest: an address field and a topics field

In particular, Custom Webhook log GraphQL objects accept aBlockLogsFilterCriteria filter which is comprised of the following objects:

- addresses: [Address!]
- Addresses accepts an array of strings which map to either a single contract address or a list of addresses from which logs should originate. If this list is empty, results will not be filtered by address.
- topics: [[Bytes32!]!]
- Array of 32 Bytes DATA topics. Topics are order-dependent. Each topic can also be an array of DATA with "or" options.

If you're not familiar with how topic filters work foreth_getLogs , here's an article to get you up to speed.

A note on specifying topic filters within Custom Webhooks:

A transaction with topics will be matched by the following topic filters:

- []
- “anything”
- [A]
- “A in first position (and anything after)”
- [[], [B]]
- “anything in first position AND B in second position (and anything after)”
- [A, B]
- “A in first position AND B in second position (and anything after)”
- [[A, B], [A, B]]
- “(A OR B) in first position AND (A OR B) in second position (and anything after)”

Specifying Addresses within Custom Webhook Event Filters

Let's investigate an example log filter together! In this snippet, we only define a single address of interest, the Bored Ape NFT contract 0xBC4CA0EdA7647A8aB7C2061c2E118A18a936f13D . In this example, we set our filter so that we return logs/events only associated with the Bored Ape NFT contract.

Get all logs emitted by the Bored Ape NFT contract in block 0x4a4d98f90439daaf082642dfbdd0f7b9a36749484582fb3d3e03bd4583da337a

{ block(hash: "0x4a4d98f90439daaf082642dfbdd0f7b9a36749484582fb3d3e03bd4583da337a") { logs(filter: {addresses: ["0xBC4CA0EdA7647A8aB7C2061c2E118A18a936f13D"], topics: []}) { topics data } } }

NOTE:

In the case where we would want apply the same filter across a list of contracts, we could simply add more addresses to the address list part of filter!

Specifying Topics within Custom Webhook Event Filters

Let's investigate an example topc filter together! In this snippet, we only define a single topic of interest, thetransfer event with a kek hash of 0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef . In this example, we set our filter so that we return logs/events only associated with the transfer event

Get all logs associated with a transfer event in block 0x4a4d98f90439daaf082642dfbdd0f7b9a36749484582fb3d3e03bd4583da337a

{ block(hash: "0x4a4d98f90439daaf082642dfbdd0f7b9a36749484582fb3d3e03bd4583da337a") { logs(filter: {addresses: [], topics:

["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef"]}) { topics data } } }

External Transaction Filters

Custom Webhook external transaction filters allow developers to filter on the to and from addresses associated with each external transaction!

What is an external transaction?

An external transaction refers to a transaction that is initiated by an external account on the Ethereum network. It involves sending a request to modify the state of the Ethereum blockchain, such as transferring Ether (the native cryptocurrency of Ethereum) or interacting with a smart contract. In particular, Custom Webhook log GraphQL objects accept aBlockTransactionsFilterCriteria filter which is comprised of the following fields!

- from: [Address!]
- The sending party of a transaction
- to: [Address!]
- The receiving party of a transaction

A transaction from theBlockTransactionsFilterCriteria will match if it has a from address that is in the list of from addresses AND a to address that is in the list of to addresses provided. Missing or empty lists will be treated as wildcards and will return a match.

Specifying

from addresses within Custom Webhook Transaction Filters

Let's investigate an example transaction filter together! In this snippet, we only define a single from address of interest 0xc83dad6e38BF7F2d79f2a51dd3C4bE3f530965D6 so that Alchemy sends out webhook emissions only associated with 0xc83dad6e38BF7F2d79f2a51dd3C4bE3f530965D6

Filter for transactions originating from 0xc83dad6e38BF7F2d79f2a51dd3C4bE3f530965D6

{ block(hash: "0x49c8fd41516ed8d8ba871c0dadaddb0928d7d03d4de569b2537343080f48c618") { hash number timestamp transactions(filter: {addresses: [{from: ["0xc83dad6e38BF7F2d79f2a51dd3C4bE3f530965D6"]}]}) { from { address }, to { address } hash, value gas status } } }

Specifying

to within Custom Webhook Transaction Filters

Let's investigate an example transaction filter together! In this snippet, we only define a single to address 0x388C818CA8B9251b393131C08a736A67ccB19297 of interest so that Alchemy sends out webhook emissions only associated with transactions sent to 0x388C818CA8B9251b393131C08a736A67ccB19297

Filter for transactions sent to 0x388C818CA8B9251b393131C08a736A67ccB19297

{ block(hash: "0x49c8fd41516ed8d8ba871c0dadaddb0928d7d03d4de569b2537343080f48c618") { hash number timestamp transactions(filter: {addresses: [{to: ["0x388C818CA8B9251b393131C08a736A67ccB19297"]}]}) { from { address }, to { address } hash, value gas status } } }

Specifying

from & to within Custom Webhook Transaction Filters

Let's investigate an example transaction filter together! In this snippet, we only define a single to address 0xeb83e695adcac2e83f290d2d2815fc58e6491d7a and a from address 0x29469395eaf6f95920e59f858042f0e28d98a20b so that Alchemy sends out webhook emissions only associated with that from and to configuration.

{ block(hash: "0x49c8fd41516ed8d8ba871c0dadaddb0928d7d03d4de569b2537343080f48c618") { hash number timestamp transactions(filter: {addresses: [{from: ["0xeb83e695adcac2e83f290d2d2815fc58e6491d7a"]}, {to: ["0x29469395eaf6f95920e59f858042f0e28d98a20b"]}]}) { from { address }, to { address } hash, value gas status } } }

Internal Transaction (Debug trace calls) Filters (BETA)

What is an internal transaction?

An internal transaction refers to a transaction that is triggered as a result of the execution of a smart contract. Unlike external transactions that are initiated by external accounts, internal transactions occur within the context of a specific smart contract and are not directly initiated by external users or applications. Similar to external transactions, Custom Webhook log GraphQL objects accept aBlockCallTracesFilterCriteria filter which is comprised of the following fields!

- from: [Address!]
- The sending party of a transaction
- to: [Address!]
- The receiving party of a transaction

A transaction from theBlockCallTracesFilterCriteria will match if it has a from address that is in the list of from addresses AND a to address that is in the list of to addresses provided. Missing or empty lists will be treated as wildcards and will return a match.

Specifying

from addresses within Custom Webhook Internal Transaction Filters

Filter for internal transactions originating from 0x5c43B1eD97e52d009611D89b74fA829FE4ac56b1

{ block { number callTracerTraces (filter: {addresses: [{from: ["0x5c43B1eD97e52d009611D89b74fA829FE4ac56b1"], to: []}]}){ from { address } to { address } type input output value gas gasUsed input output error revertReason subtraceCount traceAddressPath } } }

Specifying

to addresses within Custom Webhook Internal Transaction Filters

Filter for internal transactions going to 0x5c43B1eD97e52d009611D89b74fA829FE4ac56b1

{ block { number callTracerTraces (filter: {addresses: [{from: [], to: ["0x5c43B1eD97e52d009611D89b74fA829FE4ac56b1"]}]}){ from { address } to { address } type input output value gas gasUsed input output error revertReason subtraceCount traceAddressPath } } }

Specifying

from & to within Custom Webhook Internal Transaction Filters

{ block { number callTracerTraces (filter: {addresses: [{from: ["0x29469395eaf6f95920e59f858042f0e28d98a20b"], to: ["0x5c43B1eD97e52d009611D89b74fA829FE4ac56b1"]}]}){ from { address } to { address } type input output value gas gasUsed input output error revertReason subtraceCount traceAddressPath } } }

Updated 5 months ago

Custom Webhooks GraphQL Examples Custom Webhook Variables Did this page help you?Yes No