## 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

## Transfers API (Tx History)



## Transaction Receipts API



## Token API

# 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

## Alchemy 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

- 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

## Optimism

- 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

# eth_call - Arbitrum

post https://{network}.g.alchemy.com/v2/{apiKey} Executes a new message call immediately without creating a transaction on the block chain.

All CSS

/ *dont_have_api_sec_start* / .api_key_instruct_ban{ background: #F5FCFF; border: 1px solid rgba(207, 217, 240, 0.2); border-radius: 12px; -webkit-border-radius: 12px; display: flex; flex-wrap: wrap; padding: 33px; } .markdown-body h3 { color: #00000 !important; } .api_key_instruct_ban_lft h3{ font-size: 24px; line-height: 1.3; letter-spacing: -0.03em; font-weight: 700; font-family: 'Inter', sans-serif; margin-bottom: 7px; margin-top: 0px; color: #00000 !important; } .api_key_instruct_ban_lft h3:last-child{ margin-bottom: 0; } .api_key_instruct_ban_lft p{ font-size: 14px; line-height: 1.3; color: #000000; font-family: 'Inter', sans-serif; font-weight: 400; } .gt_startd_vbtn{ display: inline-block; color: #FFFFFF !important; background: linear-gradient(126.33deg, #36BEFF 5.38%, #733FF1 108.32%); border-radius: 6px; font-size: 16px; line-height: 1.3; font-weight: 600; font-family: 'Inter', sans-serif; padding: 9px 16px; } .gt_startd_vbtn:hover{ background: #000; color: #fff; } .api_key_instruct_ban_lft{ flex-basis: 60%; max-width: 60%; padding-right: 15px; } .api_key_instruct_ban_rtt{ flex-basis: 40%; max-width: 40%; padding-left: 15px; align-self: center; } / *dont_have_api_sec_end* /

```
/* ========== responsive css ========== */

@media(min-width:1025px) {}

@media(max-width:1199px) {
    .api_main {
        max-width: 100%;
    }
    .api_main_cont ul li a {
        width: 100%;
    }
    .left_icon .evm_part {
        width: 73%;
        margin: 22px auto auto auto;
    }
    .api_main_cont ul li a {
        padding: 10px 18px;
        border-radius: 12px;
    }
    .api_main_cont ul li a:hover::before {
        border-radius: 12px;
    }
    .learn_box,
    .lear_outer {
        height: 100%;
    }
    .navbar-nav>li>a {
        font-size: 15px;
        padding: 8px 10px;
    }
    .top_header_links ul li a{
        font-size: 15px;
    }
    .footer_links_box ul li a{
        font-size: 13px;
```

```
    }
    /* dont_have_api_sec_start */
    .api_key_instruct_ban{
            display: block;
            text-align: center;
            background: linear-gradient(180deg, #EBF9FF 0%, #EEF3FE 100%);
        }
        .api_key_instruct_ban_lft,.api_key_instruct_ban_rtt{
            max-width: 100%;
            padding: 0;
            flex-basis: 100%;
        }
        .api_key_instruct_ban_lft{
            margin-bottom: 30px;
        }
    /* dont_have_api_sec_end */
}

@media (max-width: 768px) {
        /* dont_have_api_sec_start */

        .api_key_instruct_ban_lft h3{
            font-size: 30px;
        }
        .api_key_instruct_ban_lft p{
            font-size: 16px;
        }
        .wrapper_body_cmn_out{
            max-width: 100%;
        }
        .api_key_instruct_ban_lft h3 {
            font-size: 40px;
            margin-bottom: 16px;
        }

        /* dont_have_api_sec_end */
}

@media (max-width: 350px){
    /* dont_have_api_sec_start  */
    .api_key_instruct_ban_lft h3{
        font-size: 36px;
    }
    .api_key_instruct_ban{
        padding: 30px;
    }
    /* dont_have_api_sec_end */
}
```

whole_Section_wrapperdont_have_api_sec_start### Don’t have an API key?

Start using this API in your app today.Get started for free dont_have_api_sec_end This is one of the most commonly used API calls. It is used to read from the blockchain which includes executing smart contracts but does not publish anything to the blockchain. This call does not consume any Ether.

We can call any function of a smart contract using the eth_call method and it returns us any data that the function returns (in hex format). For read-only functions, it returns what the read-only function returns. For functions that change the state of the contract, it executes that transaction locally and returns any data returned by the function.

Calling the read-only function is a common use case because all read-only functions return something that we can read using this method.

## Use cases for

eth_call

eth_call is used to call read-only functions of a smart contract. For example, calling the balanceOf function of an ERC20 token contract.

- How to Get ERC-20 Token Balance at a Given Block
- How to decode an eth_call response

Starting fromGeth 1.9.13 , eth_call will check the balance of the sender (to make sure that the sender has enough gas to complete the request) before executing the call when one of the following conditions is true:

1. the
2. gas_price
3. parameter is populated, or
4. the contract function being called (i.e. in
5. data
6. modifies blockchain state)\

In these two cases, even though the eth_call requests don't consume any gas, the from address must have enough gas to execute the call as if it were a write transaction because eth_call is being used to simulate the transaction.

## Parameters

1. Object
2. 
    - The transaction call object
3. 
    - from
4. 
    - :
5. 
    - DATA
6. 
    - , 20 Bytes - (optional) The address the transaction is sent from.
7. 
    - to
8. 
    - :
9. 
    - DATA
10. 
    - , 20 Bytes - The address the transaction is directed to.
11. 
    - gas
12. 
    - :
13. 
    - QUANTITY
14. 
    - 
        - (optional) Integer of the gas provided for the transaction execution.
15. 
    - eth_call
16. 
    - consumes zero gas, but this parameter may be needed by some executions.NOTE: this parameter has a cap of550 million
17. 
    - gas per request. Reach out to us atsupport@alchemy.com
18. 
    - if you want to increase this limit!
19. 
    - gasPrice
20. 
    - :
21. 
    - QUANTITY
22. 
    - 
        - (optional) Integer of the
23. 
    - gasPrice
24. 
    - used for each paid gas.
25. 
    - value
26. 
    - :
27. 
    - QUANTITY
28. 
    - 
        - (optional) Integer of the value sent with this transaction
29. 
    - data
30. 
    - :
31. 
    - DATA
32. 
    - 
        - (optional) Hash of the method signature and encoded parameters. For details seeEthereum Contract ABI
33. QUANTITY|TAG
34. 
    - integer block number, or the string "latest", "earliest" or "pending" (see thedefault block parameter
35. ), OR the
36. blockHash
37. (in accordance withEIP-1898
38. )NOTE: the parameter is an object instead of a string and should be specified as:
39. {"blockHash": "0x"}.
40. Learn morehere
41. .
42. Object
43. 

- State override set

44. The State Override Set option allows you to change the state of a contract before executing the call. This means you can modify the values of variables stored in the contract, such as balances and approvals for that call without actually modifying the contract on the blockchain.
45. 
46. In more technical terms, the state override set is an optional parameter that allows executing the call against a modified chain state. It is an address-to-state mapping, where each entry specifies some state to be overridden prior to executing the call. Each address maps to an object containing:

FIELD TYPE BYTES DESCRIPTION balance Quantity <32 Fake balance to set for the account before executing the call. nonce Quantity <8 Fake nonce to set for the account before executing the call. code Binary any Fake EVM bytecode to inject into the account before executing the call. state Object any Fake key-value mapping to override all slots in the account storage before executing the call. stateDiff Object any Fake key-value mapping to override individual slots in the account storage before executing the call.

Override Example :

Here's a simple code snippet in JavaScript that shows how you can use the State Override Set to mock an approval for a token transfer:

Override Example // Import the ethers.js library const ethers = require ("ethers" );// The address of the DAI token contract const dai = "0x6b175474e89094c44da98b954eedeac495271d0f" ;// The address of the sender const fromAddr = "0xde0B295669a9FD93d5F28D9Ec85E40f4cb697BAe" ;// The address of the recipient const toAddr = "0x52bc44d5378309ee2abf1539bf71de1b7d7be3b5" ;// The allowance slot on the DAI contract (this may differ from contract to contract) const slot = 3 ;// Use the solidityKeccak256 function from the ethers.js library to calculate the index for the allowance mapping const temp = ethers .utils .solidityKeccak256 ( ["uint256" ,"uint256" ], [fromAddr ,slot ] );const index = ethers .utils .solidityKeccak256 ( ["uint256" ,"uint256" ], [toAddr ,temp ] );// The stateDiff object to mock an approval const stateDiff = {

: { [index ]:ethers .constants .MaxUint256 .toHexString (),// setting the allowance to the max value of uint256 }, }, };// Create an instance of the Ethereum provider const provider = new ethers .providers .JsonRpcProvider ("Your-Alchemy-API-URL" );// The parameters for the eth_call method const callParams = [ {to :dai ,data :"0xdd62ed3e..." // The method signature and arguments for the call },"latest" ];// Call the contract method without state overrides const call1 = await provider .send ("eth_call" ,callParams );// Call the contract method with state overrides const call2 = await provider .send ("eth_call" , [... callParams ,stateDiff ]);// Log the results of both calls console .log (call1 );console .log (call2 ); Code Explanation :

- We first import the
- ethers.js
- library, which provides a convenient set of tools for working with EVM chains.
- Next, we define the address of the DAI token contract and the addresses of the sender and recipient.
- We then calculate the index for the allowance mapping in the token contract. This involves using the
- solidityKeccak256
- function from the
- ethers.js
- library to calculate a unique identifier for the mapping based on the sender and recipient addresses.
- The
- stateDiff
- object is created to mock an approval, which is done by setting the state of the index in the allowance mapping to the maximum possible value (
- ethers.constants.MaxUint256
- ).
- An instance of the Ethereum provider is created. This provider will be used to make calls to the Ethereum network.
- The
- callParams
- constant is created that specifies the parameters for the
- eth_call
- method.
- The contract method is called without state overrides and the result is stored in the
- call1
- constant.
- The contract method is called with state overrides and the result is stored in the
- call2
- constant.
- The results of both calls are logged to the console.

The State Override option is mainly used for testing purposes, as it allows developers to temporarily modify the state of the chain to simulate specific scenarios and test the behavior of smart contracts.

Note

eth_call has a timeout restriction at the node level. Batching multiple eth_call together on-chain using pre-deployed smart contracts might result in unexpected timeouts that cause none of your calls to complete. Instead, consider serializing these calls, or using smaller batches if they fail with a node error code. JavaScript params : [ {"from" :"0xb60e8dd61c5d32be8058bb8eb970870f07233155" ,"to" :"0xd46e8dd67c5d32be8058bb8eb970870f07244567" ,"gas" :"0x76c0" ,"gasPrice" :"0x9184e72a000" ,"value" :"0x9184e72a" ,"data" :"0xd46e8dd67c5d32be8d46e8dd67c5d32be8058bb8eb970870f072445675058bb8eb970870f072445675" },"latest" ]

## Returns

DATA - the return value of the executed contract.

Example

Request

SDK ethers.js web3.py cURL Postman // Setup: npm install alchemy-sdk // Github: https://github.com/alchemyplatform/alchemy-sdk-js import {Network ,Alchemy }from "alchemy-sdk" ;// Optional config object, but defaults to demo api-key and eth-mainnet. const settings = {apiKey :"demo" ,// Replace with your Alchemy API Key. network :Network .ETH_MAINNET ,// Replace with your network. };const alchemy = new Alchemy (settings );// Make a sample eth_call alchemy .core .call ({to :"0x4976fb03C32e5B8cfe2b6cCB31c09Ba78EBaBa41" ,gas :"0x76c0" ,gasPrice :"0x9184e72a000" ,data :"0x3b3b57debf074faa138b72c65adbdcfb329847e4f2c04bde7f7dd7fcad5a52d2f395a558" , }) .then (console .log ); // Installation instructions: https://docs.ethers.io/v5/getting-started/#installing async function main () {const {ethers }= require ("ethers" );// Replace with your Alchemy API key: const apiKey = "demo" ;// Initialize an ethers instance const provider = new ethers .providers .AlchemyProvider ("homestead" ,apiKey );// Query the blockchain (replace example parameters) const data = await provider .call ( {"from" :"0xb60e8dd61c5d32be8058bb8eb970870f07233155" ,"to" :"0xd46e8dd67c5d32be8058bb8eb970870f07244567" ,"gas" :"0x76c0" ,"gasPrice" :"0x9184e72a000" ,"value" :"0x9184e72a" ,"data" :"0xd46e8dd67c5d32be8d46e8dd67c5d32be8058bb8eb970870f072445675058bb8eb970870f072445675" },"latest" );// Print the output to console console .log (data ); }main ()

# Installation Instructions: https://web3py.readthedocs.io/en/latest/quickstart.html#installation

from web3 import Web3 ,HTTPProvider

# Replace with your Alchemy API key:

# apiKey

"demo"

# Initialize a Web3.py instance

# web3

Web3 (Web3 .HTTPProvider ('https://eth-mainnet.g.alchemy.com/v2/' + apiKey ))# Query the blockchain (replace example parameters) data = web3 .eth .call ({'value' :0 ,'gas' :21736 ,'maxFeePerGas' :2000000000 ,'maxPriorityFeePerGas' :1000000000 ,'to' :'0xc305c90' ,'data' :'0x477a5c98' })# Print the output to console print (data ) curl https://eth-mainnet.g.alchemy.com/v2/your-api-key -X POST -H "Content-Type: application/json" -d '{"jsonrpc":"2.0","method":"eth_call","params":[{"from": "0xb60e8dd61c5d32be8058bb8eb970870f07233155","to": "0xd46e8dd67c5d32be8058bb8eb970870f07244567","gas": "0x76c0","gasPrice": "0x9184e72a000","value": "0x9184e72a","data": "0xd46e8dd67c5d32be8d46e8dd67c5d32be8058bb8eb970870f072445675058bb8eb970870f072445675"}, "latest"],"id":1}' URL: https://eth-mainnet.g.alchemy.com/v2/your-api-key RequestType: POST Body:

{ "jsonrpc":"2.0", "method":"eth_call", "params":[{"from": "0xb60e8dd61c5d32be8058bb8eb970870f07233155","to": "0xd46e8dd67c5d32be8058bb8eb970870f07244567","gas": "0x76c0","gasPrice": "0x9184e72a000","value": "0x9184e72a","data": "0xd46e8dd67c5d32be8d46e8dd67c5d32be8058bb8eb970870f072445675058bb8eb970870f072445675"}, "latest"], "id":1 }

### Result

JavaScript {"jsonrpc" :"2.0" ,"id" :1 ,"result" :"0x" }

Path Params apiKey string required .custom-style { color: #048FF4; } For higher throughput,create your own API key Body Params Accepts the transaction call object, state overrides and the block number / block hash / block tag to execute the call on. id integer jsonrpc string method string params array params Transaction Object Block Number, Tag, or Hash Transaction Object Block Number, Tag, or Hash

Response

# 200

The result of the call.

Response body object id integer jsonrpc string result string The result of the call

Updated 4 months ago

Arbitrum API Endpoints eth_estimateGas - Arbitrum Did this page help you?Yes No