

# eth\_getProof

Returns the account and storage values, including the Merkle proof, of the specified account.

## Parameters

- address
  - : A string representing the address (20 bytes) to check for balance.
- storageKeys
  - : An array of 32-byte storage keys to be proofed and included.
- blockParameter
  - : A hexadecimal block number, or one of the string tags latest
  - , earliest
  - , pending
  - , safe
  - , or finalized
  - . See the [default block parameter](#)
  - .

## Returns

- balance
  - : Hexadecimal of the current balance in wei.
- codeHash
  - : The 32-byte hash of the code of the account.
- nonce
  - : The nonce of the account.
- storageHash
  - : 32 bytes. The SHA3 of the StorageRoot. All storage will deliver a Merkle proof starting with this rootHash
  - .
- accountProof
  - : An array of RLP-serialized MerkleTree-Nodes, starting with the stateRoot-Node, following the path of the SHA3 (address) as key.
- storageProof
  - : An array of storage-entries as requested. Each entry is an object with these properties: \* key
    - : The requested storage key.
  - value
    - : The storage value.
  - proof
    - : An array of RLP-serialized MerkleTree-Nodes, starting with the storageHash-Node, following the path of the SHA3 (key) as path.

## Example

Replace YOUR-API-KEY with an API key from your [Infura dashboard](#) .

## Request

- cURL
- WSS

```
curl https://optimism-mainnet.infura.io/v3/YOUR-API-KEY \-X POST \-H "Content-Type: application/json" \-d '{"jsonrpc": "2.0", "method": "eth_getProof", "id": 1, "params": ["0x7F0d15C7FAae65896648C8273B6d7E43f58Fa842", ["0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cad001622fb5e363b421"], "latest"]}' wscat -c wss://optimism-mainnet.infura.io/ws/v3/YOUR-API-KEY -x '{"jsonrpc": "2.0", "method": "eth_getProof", "params": ["0x7F0d15C7FAae65896648C8273B6d7E43f58Fa842", ["0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cad001622fb5e363b421"], "latest"], "id": 1}'
```

## Response

- JSON

```
{ "id": 1, "jsonrpc": "2.0", "result": { "accountProof": [ "0xf90211a...0701bc80", "0xf90211a...0d832380",  
"0xf90211a...5fb20c80", "0xf90211a...0675b80", "0xf90151a0...ca08080" ], "address" :  
"0x7f0d15c7faae65896648c8273b6d7e43f58fa842", "balance" : "0x0", "codeHash" :  
"0xc5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470", "nonce" : "0x0", "storageHash" :  
"0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cad001622fb5e363b421", "storageProof" : [ { "key" :  
"0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cad001622fb5e363b421", "proof" : [], "value" : "0x0" } ] } }
```

Last updated on Apr 19, 2024 [Previous eth\\_getLogs](#) [Next eth\\_getStorageAt](#)