
title: Block explorers description: An introduction to block explorers, your portal into the world of blockchain data, where you can query information about transactions, accounts, contracts, and more. lang: en sidebarDepth: 3

Block explorers are your portal to Ethereum's data. You can use them to see real-time data on blocks, transactions, miners, accounts, and other on-chain activity.

Prerequisites {#prerequisites}

You should understand the basic concepts of Ethereum so you can make sense of the data that a block explorer gives you. Start with [an intro to Ethereum](#).

Services {#services}

- [Etherscan](#) -Also available in Chinese, Korean, Russian, and Japanese
- [Beaconcha.in](#)
- [Blockchair](#) -Also available in Spanish, French, Italian, Dutch, Portuguese, Russian, Chinese, and Farsi
- [Blockscout](#)
- [Etherchain](#)
- [Ethplorer](#) -Also available in Chinese, Spanish, French, Turkish, Russian, Korean and Vietnamese
- [OKLink](#)
- [Otterscan](#)
- [Rantom](#)
- [Sirato](#)
- [EthVM](#)
- [DexGuru Block Explorer](#)

Data {#data}

Ethereum is transparent by design so everything is verifiable. Block explorers provide an interface for getting this information. And this is for both the main Ethereum network and the testnets, should you need that data. Data is divided into execution data and consensus data. The execution data refers to the transactions that have been executed in a specific block. The consensus data refers to the blocks themselves and the validators who proposed them.

Here's a summary of the types of data you can get from a block explorer.

Execution data {#execution-data}

New blocks are added to Ethereum every 12 seconds (unless a block proposer misses its turn), so a near-constant stream of data gets added to block explorers. Blocks contain a lot of important data that you may find useful:

Standard data

- Block height - The block number and length of the blockchain (in blocks) on creation of the current block
- Timestamp - The time at which a block was proposed
- Transactions - The number of transactions included within the block
- Fee recipient - The address that received gas fee tips from transactions
- Block Reward - The amount of ETH awarded to the validator who proposed the block
- Size - The size of the data within the block (measured in bytes)
- Gas used - The total units of gas used by the transactions in the block
- Gas limit - The total gas limits set by the transactions in the block
- Base fee per gas - The minimum multiplier required for a transaction to be included in a block
- Burnt fees - How much ETH is burned in the block
- Extra data - Any extra data the miner has included in the block

Advanced data

- Hash - The cryptographic hash that represents the block header (the unique identifier of the block)
- Parent hash - The hash of the block that came before the current block
- StateRoot - The root hash of Merkle trie which stores the entire state of the system

Gas {#gas}

Not only will block explorers give you data about Gas usage in transactions and blocks, but some will give you information on the network's current gas prices. This will help you understand network usage, submit safe transactions and not overspend on gas. Look out for APIs that can help you get this information into your product's interface. Gas-specific data covers:

- Estimated units of gas needed for a safe but slow transaction (+ estimated price and duration)
- Estimated units of gas needed for an average transaction (+ estimated price and duration)
- Estimated units of gas needed for a fast transaction (+ estimated price and duration)
- Average confirmation time based on gas price
- Contracts that are consuming gas - in other words, popular products that are seeing lots of usage on the network
- Accounts that are spending gas - in other words, frequent network users

Transactions {#transactions}

Block explorers have become a common place for people to track the progress of their transactions. That's because the level of detail you can get provides extra certainty. Transaction data includes:

Standard data

- Transaction hash - A hash generated when the transaction is submitted
- Status - An indication of whether the transaction is pending, failed or a success
- Block - The block in which the transaction has been included
- Timestamp - The time at which a miner mined the transaction
- From - The address of the account that submitted the transaction
- To - The address of the recipient or smart contract that the transaction interacts with
- Tokens transferred - A list of tokens that were transferred as part of the transaction
- Value - The total ETH value being transferred
- Transaction fee - The amount paid to the miner to process the transaction (calculated by gas price*gas used)

Advanced data

- Gas limit - The maximum numbers of gas units this transaction can consume
- Gas used - The actual amount of gas units the transaction consumed
- Gas price - The price set per gas unit
- Nonce - The transaction number for the `from` address (bear in mind this starts at 0 so a nonce of `100` would actually be the 101st transaction submitted by this account)
- Input data - Any extra information required by the transaction

Accounts {#accounts}

There's a lot of data that you can access about an account. This is why it's often recommended to use multiple accounts so that your assets and value can't be easily tracked. There are also some solutions being developed to make transactions and account activity more private. But here's the data that's available for accounts:

User accounts

- Account address - The public address you can use to send funds to
- ETH balance - The amount of ETH associated with that account
- Total ETH value - The value of the ETH
- Tokens - The tokens associated with the account and their value

- Transaction history - A list of all the transactions where this account was either the sender or the recipient

Smart contracts

Smart contract accounts have all the data that a user account will have, but some block explorers will even display some code information too. Examples include:

- Contract creator - The address that deployed the contract to Mainnet
- Creation transaction - The transaction that included the deployment to Mainnet
- Source code - The solidity or vyper code of the smart contract
- Contract ABI - The Application Binary Interface of the contract—the calls the contract makes and the data received
- Contract creation code - The compiled bytecode of the smart contract—created when you compile a smart contract written in Solidity or Vyper, etc.
- Contract events - A history of the methods called in the smart contract—basically a way to see how the contract is being used and how often

Tokens {#tokens}

Tokens are a type of contract so they'll have similar data to a smart contract. But because they have value and can be traded they have additional data points:

- Type - Whether they're an ERC-20, ERC-721 or another token standard
- Price - If they're an ERC-20 they'll have a current market value
- Market cap - If they're an ERC-20 they'll have a market cap (calculated by price*total supply)
- Total supply - The number of tokens in circulation
- Holders - The number of addresses that hold the token
- Transfers - The number of times the token has been transferred between accounts
- Transaction history - A history of all the transactions including the token
- Contract address - The address of the token that was deployed to Mainnet
- Decimals - ERC-20 tokens are divisible and have decimal places

Network {#network}

Some block data is concerned about the health of Ethereum more holistically.

- Total transactions - The number of transactions since Ethereum was created
- Transactions per second - The number of transactions processable within a second
- ETH price - The current valuations of 1 ETH
- Total ETH supply - Number of ETH in circulation—remember new ETH is created with the creation of every block in the form of block rewards
- Market cap - Calculation of price*supply

Consensus layer data {#consensus-layer-data}

Epoch {#epoch}

For security reasons, randomized committees of validators are created at the end of every epoch (every 6.4 minutes). Epoch data includes:

- Epoch number
- Finalized status - Whether the epoch has been finalized (Yes/No)
- Time - The time the epoch ended
- Attestations - The number of attestations in the epoch (votes for blocks within slots)
- Deposits - The number of ETH deposits included in the epoch (validators must stake ETH to become validators)
- Slashings - Number of penalties given to proposers of blocks or attestors
- Voting participation - The amount of staked ETH used to attest blocks
- Validators - Number of validators active for the epoch

- Average Validator balance - Average balance for active validators
- Slots - Number of slots included in the epoch (slots include one valid block)

Slot {#slot}

Slots are opportunities for block creation, the data available for each slot includes:

- Epoch - The epoch in which the slot is valid
- Slot number
- Status - The status of the slot (Proposed/Missed)
- Time - The slot timestamp
- Proposer - The validator that proposed the block for the slot
- Block root - The hash-tree-root of the BeaconBlock
- Parent root - The hash of the block that came before
- State root - The hash-tree-root of the BeaconState
- Signature
- Randao reveal
- Graffiti - A block proposer can include 32 byte long message to its block proposal
- Execution Data
- Block hash
- Deposit count
- Deposit root
- Attestations - Number of attestations for the block in this slot
- Deposits - The number of deposits during this slot
- Voluntary exits - The number of validators that left during the slot
- Slashings - Number of penalties given to proposers of blocks or attestors
- Votes - The validators that voted for the block in this slot

Blocks {#blocks-1}

Proof-of-stake divides time into slots and epochs. So that means new data!

- Proposer - The validator that was algorithmically chosen to propose the new block
- Epoch - The epoch in which the block was proposed
- Slot - The slot in which the block was proposed
- Attestations - The number of attestation included in the slot—attestations are like votes that indicate the block is ready to go to the Beacon Chain

Validators {#validators}

Validators are responsible for proposing blocks and attesting to them within slots.

- Validator number - Unique number that represents the validator
- Current balance - The validator's balance including rewards
- Effective balance - The validator's balance that is used for staking
- Income - The rewards or penalties received by the validator
- Status - Whether the validator is currently online and active or not
- Attestation effectiveness - The average time it takes for the validator's attestations to be included in the chain
- Eligibility for activation - Date (and epoch) when the validator became available to validate
- Active since - Date (and epoch) when the validator became active
- Proposed blocks - The block that the validator has proposed
- Attestations - The attestations that the validator has provided
- Deposits - The from address, transaction hash, block number, timestamp, amount and status of the staking deposit made by the validator

Attestations {#attestations}

Attestations are "yes" votes to include blocks in the chain. Their data relates to a record of the attestation and the validators who attested

- Slot - The slot in which the attestation took place
- Committee index - The index of the committee at the given slot
- Aggregation bits - Represents the aggregated attestation of all participating validators in the attestation
- Validators - The validators that provided attestations
- Beacon block root - Points to the block to which validators are attesting
- Source - Points to the latest justified epoch
- Target - Points to the latest epoch boundary
- Signature

Network {#network-1}

The consensus layer top-level data includes the following:

- Current epoch
- Current slot
- Active validators - Number of active validators
- Pending validators - Number of validators waiting for to be made active
- Staked ETH - Amount of ETH staked in the network
- Average balance - Average ETH balance of validators

Block explorers {#block-explorers}

- [Etherscan](#) - a block explorer you can use to fetch data for Ethereum Mainnet and Goerli Testnet
- [Beaconcha.in](#) - an open source block explorer for Ethereum Mainnet and Goerli Testnet
- [Blockchair](#) - the most private Ethereum explorer. Also for sorting and filtering (mempool) data
- [Etherchain](#) - a block explorer for the Ethereum Mainnet
- [Ethplorer](#) - a block explorer with a focus on tokens for the Ethereum Mainnet and the Kovan testnet
- [Rantom](#) - A user-friendly open-source DeFi & NFT transaction viewer for detailed insights

Further reading {#further-reading}

Know of a community resource that helped you? Edit this page and add it!

Related topics {#related-topics}

- [Transactions](#)
- [Accounts](#)
- [Networks](#)