

Transaction Fees

The small fees paid to process [instructions](#) on the Solana blockchain are known as "transaction fees".

As each transaction (which contains one or more instructions) is sent through the network, it gets processed by the current leader validation-client. Once confirmed as a global state transaction, this transaction fee is paid to the network to help support the economic design of the Solana blockchain.

NOTE: Transactions fees are different from the blockchain's data storage fee called [rent](#)

Transaction Fee Calculation#

Currently, the amount of resources consumed by a transaction do not impact fees in any way. This is because the runtime imposes a small cap on the amount of resources that transaction instructions can use, not to mention that the size of transactions is limited as well. So right now, transaction fees are solely determined by the number of signatures that need to be verified in a transaction. The only limit on the number of signatures in a transaction is the max size of transaction itself. Each signature (64 bytes) in a transaction (max 1232 bytes) must reference a unique public key (32 bytes) so a single transaction could contain as many as 12 signatures (not sure why you would do that). The fee per transaction signature can be fetched with the solana cli:

solana fees Blockhash: 8eULQbYYp67o5tGF2gxACnBCKAE39TetbYYMGTX3iBFc Lamports per signature: 5000 Last valid block height: 94236543 The solana cli fees subcommand calls the `getFees` RPC API method to retrieve the above output information, so your application can call that method directly as well:

```
curl http://api.mainnet-beta.solana.com -H "Content-Type: application/json" -d '{"jsonrpc":"2.0","id":1,"method":"getFees"}'
```

Result: (lastValidSlot removed since it's inaccurate)

```
{ "jsonrpc": "2.0", "result": { "context": { "slot": 106818885 }, "value": { "blockhash": "78e3YBCMxJBIPD1HpyVtVfZzFPG6nUycnQcyNMSUQzB", "feeCalculator": { "lamportsPerSignature": 5000 }, "lastValidBlockHeight": 96137823 }, "id": 1 }
```

Fee Determinism#

It's important to keep in mind that fee rates (such as `lamports_per_signature`) are subject to change from block to block (though that hasn't happened in the full history of the mainnet-beta cluster). Despite the fact that fees can fluctuate, fees for a transaction can still be calculated deterministically when creating (and before signing) a transaction. This determinism comes from the fact that fees are applied using the rates from the block whose blockhash matches the `recent_blockhash` field in a transaction. Blockhashes can only be referenced by a transaction for a few minutes before they expire.

Transactions with expired blockhashes will be ignored and dropped by the cluster, so it's important to understand how expiration actually works. Before transactions are added to a block and during block validation, [each transaction's recent blockhash is checked](#) to ensure it hasn't expired yet. The max age of a transaction's blockhash is only 150 blocks. This means that if no slots are skipped in between, the blockhash for block 100 would be usable by transactions processed in blocks 101 to 252, inclusive (during block 101 the age of block 100 is "0" and during block 252 its age is "150"). However, it's important to remember that slots may be skipped and that age checks use "block height" not "slot height". Since slots are skipped occasionally, the actual age of a blockhash can be a bit longer than 150 slots. At the time of writing, slot times are about 500ms and skip rate is about 5% so the expected lifetime of a transaction which uses the most recent blockhash is about 1min 19s.

Fee Collection#

Transactions are required to have at least one account which has signed the transaction and is writable. Writable signer accounts are serialized first in the list of transaction accounts and the first of these accounts is always used as the "fee payer".

Before any transaction instructions are processed, the fee payer account balance will be deducted to pay for transaction fees. If the fee payer balance is not sufficient to cover transaction fees, the transaction will be dropped by the cluster. If the balance was sufficient, the fees will be deducted whether the transaction is processed successfully or not. In fact, if any of the transaction instructions return an error or violate runtime restrictions, all account changes except the transaction fee deduction will be rolled back.

Fee Distribution#

Transaction fees are partially burned and the remaining fees are collected by the validator that produced the block that the corresponding transactions were included in. The transaction fee burn rate was initialized as 50% when inflation rewards were enabled at the beginning of 2021 and has not changed so far. These fees incentivize a validator to process as many transactions as possible during its slots in the leader schedule. Collected fees are deposited in the validator's account (listed in the leader schedule for the current slot) after processing all of the transactions included in a block.

