

# Anatomy of a Transaction

A transaction is a request from a user to the network to perform a set of actions. To create a transaction, the user must specify the following fields:

- Signer
- : the account that signs the transaction
- Actions
- : the set of actions to be performed (see below)
- Receiver
- : the account on which to perform the actions

In addition, a transaction has the following fields to ensure its integrity and validity:

- PublicKey
- : the public key used to sign the Transaction (so the network can verify the signature)
- Nonce
- : a number that is incremented for each transaction sent by the Signer
- BlockHash
- : the hash of a recent block, to limit the time-validity of the transaction

Users create transactions and sign them with their private keys. Then, the transaction and its signature are broadcasted together to the network, where they are validated and processed.

tip Each transaction has exactly one Signer and Receiver, but can have multiple Actions

## Actions

Each transaction can have one or multiple Actions, which are the actual operations to be performed on the Receiver account. There are 9 types of actions that can be performed:

1. FunctionCall
2. : to invoke a function on a contract (optionally attaching NEAR to the call)
3. Transfer
4. : to transfer tokens to another accounts
5. DeployContract
6. : to deploy a contract in the account
7. CreateAccount
8. : to create a new sub-account (e.g. ana.near
9. can create sub.ana.near
10. )
11. DeleteAccount
12. : to delete the account (transferring the remaining balance to a beneficiary)
13. AddKey
14. : to add a new key to the account (either FullAccess
15. or FunctionCall
16. access)
17. DeleteKey
18. : to delete an existing key from the account
19. DelegateActions
20. : to create a meta-transaction
21. Stake
22. : special action to express interest in becoming a network validator

For example, bob.near can bundle the following actions in a single transaction:

- Create the account contract.bob.near
- Transfer 5 NEAR to contract.bob.near
- Deploy a contract in contract.bob.near
- Call the function init
- in contract.bob.near

The Actions are executed in the order they are specified in the transaction. If any of the actions fails, the transaction is discarded and none of the actions take effect.

One Receiver Notice that all actions are performed on the same account. It is not possible to perform actions on multiple accounts in a single transaction, because transactions can have only one receiver [Edit this page](#) Last updated on Mar 25, 2024 by gagdiez Was this page helpful? Yes No

