Core Concepts

Build a strong understanding of the core concepts that make Solana different from other blockchains. Understanding the "Solana programming model" through these core concepts is very important to maximize your success as a Solana blockchain developer.

Solana Account Model#

On Solana, all data is stored in what are referred to as "accounts". The way data is organized on the Solana blockchain resembles akey-value store , where each entry in the database is called an "account".

Learn more aboutAccounts here.

Transactions and Instructions#

On Solana, we sendtransactions to interact with the network. Transactions include one or moreinstructions , each representing a specific operation to be processed. The execution logic for instructions is stored onprograms deployed to the Solana network, where each program stores its own set of instructions.

Learn more aboutTransactions andInstructions here.

Fees on Solana#

The Solana blockchain has a few different types of fees and costs that are incurred to use the permissionless network. These can be segmented into a few specific types:

- Transaction Fees
-
    - A fee to have
- validators process transactions/instructions
- Prioritization Fees
-
    - An optional
- fee to boost transactions processing order
- Rent
-
    - A withheld balance to keep data stored
- on-chain

Learn more aboutFees on Solana here.

Programs on Solana#

In the Solana ecosystem, "smart contracts" are called programs. Each program is an on-chain account that stores executable logic, organized into specific functions referred to asinstructions and called viainstruction handler functions within the respective deployed program.

Learn more aboutPrograms on Solana here.

Program Derived Address#

Program Derived Addresses (PDAs) provide developers on Solana with two main use cases:

- Deterministic Account Addresses
- : PDAs provide a mechanism to
- deterministically derive an address using a combination of optional "seeds"
- (predefined inputs) and a specific program ID.
- Enable Program Signing
- : The Solana runtime enables programs to "sign" for
- PDAs which are derived from its program ID.

You can think of PDAs as a way to create hashmap-like structures on-chain from a predefined set of inputs (e.g. strings, numbers, and other account addresses).

Learn more aboutProgram Derived Address here.

Cross Program Invocation#

A Cross Program Invocation (CPI) refers to when one program invokes the instructions of another program. This mechanism allows for the composability of Solana programs.

You can think of instructions as API endpoints that a program exposes to the network and a CPI as one API internally invoking another API.

Learn more aboutCross Program Invocation here.

Tokens on Solana#

Tokens are digital assets that represent ownership over diverse categories of assets. Tokenization enables the digitalization of property rights, serving as a fundamental component for managing both fungible and non-fungible assets.

- Fungible Tokens represent interchangeable and divisible assets of the same
- type and value (ex. USDC).
- Non-fungible Tokens (NFT) represent ownership of indivisible assets (e.g.
- artwork).

Learn more aboutTokens on Solana here.

Clusters and Endpoints#

The Solana blockchain has several different groups of validators, known asClusters . Each serving different purposes within the overall ecosystem and containing dedicated api nodes to fulfillJSON-RPC requests for their respective Cluster.

The individual nodes within a Cluster are owned and operated by third parties, with a public endpoint available for each.

There are three primary clusters on the Solana network, each with a different public endpoint:

- Mainnet -https://api.mainnet-beta.solana.com
- Devnet -https://api.devnet.solana.com
- Testnet -https://api.testnet.solana.com

Learn more aboutClusters and Endpoints here.

Previous «Wallets Next Solana Account Model»