Solana Sysvar Cluster Data

Solana exposes a variety of cluster state data to programs viasysvar accounts. These accounts are populated at known addresses published along with the account layouts in thesolana-program crate , and outlined below.

There are two ways for a program to access a sysvar.

The first is to query the sysvar at runtime via the sysvar'sget() function:

let clock = Clock::get() The following sysvars supportget :

- Clock
- EpochSchedule
- Fees
- Rent
- EpochRewards

The second is to pass the sysvar to the program as an account by including its address as one of the accounts in theInstruction and then deserializing the data during execution. Access to sysvars accounts is alwaysreadonly .

let clock_sysvar_info = next_account_info(account_info_iter)?; let clock = Clock::from_account_info(&clock_sysvar_info)?; The first method is more efficient and does not require that the sysvar account be passed to the program, or specified in theInstruction the program is processing.

Clock

The Clock sysvar contains data on cluster time, including the current slot, epoch, and estimated wall-clock Unix timestamp. It is updated every slot.

- Address:SysvarC1ock11111111111111111111111111111111
- Layout:Clock
- Fields:
  - slot
  - : the current slot
  - epoch_start_timestamp
  - : the Unix timestamp of the first slot in this epoch.
  - In the first slot of an epoch, this timestamp is identical to theunix_timestamp
  - (below).
  - epoch
  - : the current epoch
  - leader_schedule_epoch
  - : the most recent epoch for which the leader schedule
  - has already been generated
  - unix_timestamp
  - : the Unix timestamp of this slot.
- Each slot has an estimated duration based on Proof of History. But in reality,
- slots may elapse faster and slower than this estimate. As a result, the Unix
- timestamp of a slot is generated based on oracle input from voting validators.
- This timestamp is calculated as the stake-weighted median of timestamp
- estimates provided by votes, bounded by the expected time elapsed since the
- start of the epoch.
- More explicitly: for each slot, the most recent vote timestamp provided by
- each validator is used to generate a timestamp estimate for the current slot
- (the elapsed slots since the vote timestamp are assumed to be
- Bank::ns_per_slot). Each timestamp estimate is associated with the stake

- delegated to that vote account to create a distribution of timestamps by
- stake. The median timestamp is used as theunix_timestamp
- , unless the
- elapsed time since theepoch_start_timestamp
- has deviated from the expected
- elapsed time by more than 25%.

EpochSchedule

The EpochSchedule sysvar contains epoch scheduling constants that are set in genesis, and enables calculating the number of slots in a given epoch, the epoch for a given slot, etc. (Note: the epoch schedule is distinct from theleader schedule)

- Address:SysvarEpochSchedu1e11111111111111111111111
- Layout:EpochSchedule

Fees

The Fees sysvar contains the fee calculator for the current slot. It is updated every slot, based on the fee-rate governor.

- Address:SysvarFees111111111111111111111111111111111
- Layout:Fees

Instructions

The Instructions sysvar contains the serialized instructions in a Message while that Message is being processed. This allows program instructions to reference other instructions in the same transaction. Read more information oninstruction introspection .

- Address:Sysvar1nstructions1111111111111111111111
- Layout:Instructions

RecentBlockhashes

The RecentBlockhashes sysvar contains the active recent blockhashes as well as their associated fee calculators. It is updated every slot. Entries are ordered by descending block height, so the first entry holds the most recent block hash, and the last entry holds an old block hash.

- Address:SysvarRecentB1ockHashes11111111111111111111
- Layout:RecentBlockhashes

Rent

The Rent sysvar contains the rental rate. Currently, the rate is static and set in genesis. The Rent burn percentage is modified by manual feature activation.

- Address:SysvarRent111111111111111111111111111111111
- Layout:Rent

SlotHashes

The SlotHashes sysvar contains the most recent hashes of the slot's parent banks. It is updated every slot.

- Address:SysvarS1otHashes111111111111111111111111
- Layout:SlotHashes

SlotHistory

The SlotHistory sysvar contains a bitvector of slots present over the last epoch. It is updated every slot.

- Address:SysvarS1otHistory11111111111111111111111
- Layout:SlotHistory

StakeHistory

The StakeHistory sysvar contains the history of cluster-wide stake activations and de-activations per epoch. It is updated at the start of every epoch.

- Address:SysvarStakeHistory1111111111111111111111111
- Layout:StakeHistory

EpochRewards

The EpochRewards sysvar tracks the progress of epoch rewards distribution. The sysvar is created in the first block of the epoch, and lasts for several blocks while paying out the rewards. When all rewards have been distributed, the sysvar is deleted. Unlike other sysvars, which almost always exist on-chain, EpochRewards sysvar only exists during the reward period. Therefore, callingEpochRewards::get() on blocks that are outside of the reward period will return an error, i.e.UnsupportedSysvar . This can serve as a method for determining whether epoch rewards distribution has finished.

- Address:SysvarEpochRewards1111111111111111111111111
- Layout:EpochRewards

LastRestartSlot

The LastRestartSlot sysvar contains the slot number of the last restart or0 (zero) if none ever happened.

- Address:SysvarLastRestartS1ot1111111111111111111
- Layout:LastRestartSlot Previous Runtime Native Programs Next Native ZK Token Proof Program