

Tracking Jito-Solana Validators

This is for staking sites or others to detect which validators run Jito-Solana and their MEV commissions.

Jito API

Endpoint:

GET <https://kobe.mainnet.jito.network/api/v1/validators>

Response:

...

```
Copy { "validators": [ { "vote_account": "9GJmEHGom9eWo4np4L5vC6b6ri1Df2xN8KFoWixvD1Bs",
"mev_commission_bps": 800, "running_jito": true }, {
"vote_account": "J1to1yufRnoWn81KYg1XkTWzmKjnYSnmE2VY8DGUJ9Qv", "mev_commission_bps": 800,
"running_jito": true }, { "vote_account": "DsiG71AvUHUEo9rMMHqM9NAWQ6ptguRAHyot6wGzLJjx",
"mev_commission_bps": 800, "running_jito": true }, ... }
```

...

On-Chain Data

In order to determine which validators are running Jito-Solana and their configuration with purely on-chain data, you can use the following method.

Each epoch, when a Jito-Solana validator is executing its first leader slot, it will submit a transaction to create a Tip Distribution Account, which is a Program-Derived Account specific to this validator and epoch. By iterating through every active validator vote account pubkey, generating the potential Tip Distribution Account address for each, and checking which accounts have been created and whose owner is the Tip Distribution Program (confirming this is a valid PDA) you can determine which validators are running Jito. Within the Tip Distribution Account data, you can also find `mev_commission_bps` which will tell you the commission this validator is charging on MEV tips (in basis points). Some useful functions and structs:

?

...

```
Copy pubfn derive_tip_distribution_account_address( tip_distribution_program_id:&Pubkey, vote_pubkey:&Pubkey,
epoch:Epoch, )->(Pubkey,u8) { Pubkey::find_program_address( &[ TipDistributionAccount::SEED,
vote_pubkey.to_bytes().as_ref(), epoch.to_le_bytes().as_ref(), ], tip_distribution_program_id, ) }
```

...

?

...

Copy /// The account that validators register as **tip_receiver** with the tip-payment program.

[account]

[derive(Default)]

```
pub struct TipDistributionAccount { /// The validator's vote account, also the recipient of remaining lamports after /// upon
closing this account. pub validator_vote_account: Pubkey,
```

```
/// The only account authorized to upload a merkle-root for this account. pub merkle_root_upload_authority: Pubkey,
```

```
/// The merkle root used to verify user claims from this account. pub merkle_root: Option
```

```
/// Epoch for which this account was created. pub epoch_created_at: u64,
```

```
/// The commission basis points this validator charges. pub validator_commission_bps: u16,
```

```
/// The epoch (upto and including) that tip funds can be claimed. pub expires_at: u64,
```

```
/// The bump used to generate this account pub bump: u8, }
```

...

Note that lower-staked validators can take a while to get their first leader slot in an epoch, so each epoch a given validator may not create its TipDistributionAccount for several hours. Therefore we recommend tracking validators over time and merging data from the previous epoch as well as the current one.

[Previous Source](#) [Next Specification](#) Last updated 3 months ago On this page * [Jito API](#) * [On-Chain Data](#)