All You Need to Know About Solana RPCs (2023)

Introduction

As a crucial part of how decentralized applications interact with blockchain networks, RPCs are a must-know for anyone navigating this rapidly-evolving space. Whether you're focused on Solana or other blockchain platforms, this guide will arm you with essential insights into the role of RPCs in both traditional and blockchain environments.

RPC Meaning: What does RPC Stand for?

In crypto, all of the data you need is on-chain. It is completely permissionless and open for anyone to view and work with. So how do you retrieve it? With RPCs.

RPCs — while technically standing for Remote Procedure Calls — are often used as a shortform term for RPC Nodes. These are nodes that participate in the blockchain network and expose methods (called remote procedure calls) that developers can call.

RPCs vs APIs

In traditional (web2) software development, you generally fetch the data you need by working with APIs. You might use Stripe APIs for working with payments data, Twilio for SMS data, Bloomberg for financial data, Plaid for banking data, and so on.

The interesting thing to note here is that the data you're looking for is often hoarded or controlled by centralized entities.

This is where RPCs come into play. They allow permissionless data to be delivered to you in an efficient way without the need for an external provider. This is data directly from making a call to the ledger.

Conventional vs Blockchain & Crypto RPCs

In both traditional and blockchain settings, RPCs serve as a means for retrieving data, but the similarities often end there. Traditional RPCs operate over various protocols and interact with centralized databases, requiring trust in a single entity for data integrity. On the other hand, blockchain RPCs commonly use HTTP with JSON-RPC, pull data from decentralized networks, and offer higher security through data verification across multiple nodes. While traditional RPCs may restrict data access, blockchain RPCs typically offer open, public data. Additionally, blockchain RPCs, particularly in networks like Solana, are optimized for speed and throughput to meet the real-time demands of decentralized applications. Understanding these key differences can guide your interaction with RPCs in the blockchain ecosystem.

Solana RPCs

Solana Mainnet RPCs

To get data from the Solana blockchains Mainnet Beta cluster, you must make a JSON-RPC request to an RPC node participating in the network.

A typical JSON-RPC request might look like the following:

This request asks the RPC node found at https://api.mainnet-beta.solana.com for the getBlock RPC method. It knows what to do when it receives this request because it is running the software client for the Solana blockchain.

In practice, no one really works with direct JSON-RPC requests like this (although you certainly can!). Most developers prefer working with RPC clients. The most commonly used one is the Solana Web3 JS library, which can be found here.To find the full list of JSON RPC methods available on Solana, please check the official documentation.

Solana Devnet RPCs

Alternatively you can access Solana's Devnet cluster by using a Devnet RPC. Solana provides one for testing purposes that is available at https://api.devnet.solana.com.

Our RPC request would then look like:

curl https://api.devnet.solana.com -X POST -H "Content-Type: application/json" -d ' { "jsonrpc": "2.0","id":1,"method":"getBlock", "params": [430, { "encoding": "json", "maxSupportedTransactionVersion":0, "transactionDetails":"full", "rewards":false }] }

Where the only thing changing is the cluster it is on. The account we are querying may have to change if there are no

transactions or information with the account on Devnet. This is because the Devnet cluster does not share the history of Mainnet Beta

Ethereum RPCs vs Solana RPCs

For Solana to function, it needs a distributed network of nodes running its software client(s).

These nodes can be divided into two types: Validators and RPCs.

Both of these nodes have one important thing in common: they both continuously watch the blockchain and store its latest data (or state).

However, there is one key difference between them.

Validators are nodes that vote (e.g.,: "yes, this block is valid") and participate in consensus.

RPCs are nodes that don't vote — but handle requests for working with on-chain data. Note that this configuration is not commonly found in other blockchains, such as Ethereum, where the Validator and RPC nodes are often one and the same.

When you make an RPC call, you're invoking a procedure/function on a node that's participating in the blockchain. Since it's participating in the network, it knows what the latest data is and can expose it to you upon request.

Helius: Solana RPC Company

Helius provides high-end RPC access that provides many benefits to developers on Solana.

Helius RPCs are powered by the highest end hardware, ensuring that your app will perform at its best, reliably, with no drama. We pride ourselves in having the best 24/7 developer support in the ecosystem. When your system is running into an unexpected issue or you have a question about an upcoming feature, we will be there to assist in an instant.

Solana RPCs: How to Get Started with Helius

RPCs are extremely important — without them, you literally can not work with the blockchain. As a result, there are companies that specialize in perfecting the RPC experience. These machines are very heavily-utilized and it's crucial for them to perform well under high loads. A good RPC provider will be reliable, have consistent performance, and will expose metrics for the end-users to view their usage patterns over time.At Helius, we are Solana-native engineers and have made it our mission to provide the best possible RPC experience for developers of all kinds.**You can get started for free, without email sign ups or credit cards, in just two clicks here.**This was meant to be a very brief and beginner-friendly introduction to Solana RPCs. You probably already knew a good amount of this information. Next up, we'll talk about RPC optimizations, caveats, and some more advanced topics — thank you for reading!