

# How Solana RPCs Work

## What are RPCs in Crypto?

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.

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.

## How do RPCs work on Solana?

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.

## How to Work With Solana RPCs?

To get data from the Solana blockchain, 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 localhost:8899 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](#).

## How to Get Started With RPCs?

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

](<https://dev.helius.xyz/dashboard/app>).

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!