

Chain-Data Query

Learn how to connect to Filecoin RPC nodes and query Filecoin chain state and data.

Connecting to Filecoin networks via public RPC nodes

To query chain state and data on any Filecoin network, it is necessary to connect to public node providers. However, it's important to note that most public node providers offer limited access, typically allowing read-only JSON RPC calls and `MpoolPush` to send signed messages to the Filecoin networks.

To explore further details about the available public RPC providers supporting Filecoin mainnet and Calibration testnet, you can refer to the following page.

- [Filecoin mainnet RPCs](#)
- [Filecoin Calibration testnet RPCs](#)
-

Ingredients

Let's use Glif nodes as an example to demonstrate how to connect to a public Filecoin RPC node provider. Additionally, we will utilize `ethers.js` to establish the connection with the RPC nodes.

- [Glif nodes](#)
- [ethers.js](#)
-

Instructions

We will use `ethers.js` to establish a connection with the public Filecoin node provided by Glif. The following code demonstrates connecting to the Filecoin Calibration testnet as an example.

```
...  
  
Copy import { ethers } from "ethers"  
  
//The public Filecoin calibration URL const fielcoin_url = 'https://api.calibration.node.glif.io/rpc/v1'  
const provider = new ethers.JsonRpcProvider(fielcoin_url)  
  
const blockNumber = await provider.getBlockNumber() console.log("Block height: ", blockNumber)  
  
...
```

The expected output:

```
...  
  
Copy Block height: 1268350  
  
...
```

Listen to smart contract events

Since the Filecoin Virtual Machine (FVM) is EVM-compatible, we can use `ethers.js` to listen to smart contract events for specific contract actions on the Filecoin network. For instance, we can monitor ERC20 token transfer events or client `contractDealProposalCreate` events.

Ingredients

We will also use `ethers.js` to connect to the public Glif node to listen to the smart contract events.

- [Glif Nodes](#)
- [ethers.js](#)
-

Instructions

Let's consider the [wFIL contract](#), an ERC-20 token on Filecoin, as an example for listening to its transfer event. To demonstrate how to listen to smart contract events using ethers, we will use the deployed wFIL token address on the Filecoin calibration network and a simplified ABI object for the transfer event. Typically, you would have the wFIL smart contract's Application Binary Interface (ABI) defined in an `abi.json` file.

...

...

- [Glif nodes](#)
- [ethers.js](#)
-

...

...

" " "

[illegible]

'0xd388aB098ed3E84c0D808776440B48F685198498', to: '0x44061AA8Df5b33a997CE97d80c700d0C655Dc3f2', amount:
[BigNumber]] },

...

[Previous dApps](#) [Next Oracles](#)

Last updated 21 days ago