

# Customized scan module workflow

## Pre-required

Binance Smart Chain is the blockchain that runs in parallel to Binance Chain. Unless BS, BSC has the great functionality, the compatibility with EVM (Ethereum Virtual Machine). So it's possible to access the BSC node via JSON-RPC like in the Ethereum.

1. Possible to run a **Full node** to listen to and broadcast live updates on transactions, blocks, and consensus activities.

Advantage: it's the best way to run everything in your control (Very safety and fully-control)

Disadvantage: Very expensive (more time & effort & budget)

## Minimum Requirements

The hardware must meet certain requirements to run a node.

### Fullnode

- VPS running recent versions of Mac OS X or Linux.
- 1T GB of free disk space, solid-state drive(SSD).
- 8 cores of CPU and 16 gigabytes of memory (RAM).
- A broadband Internet connection with upload/download speeds of at least 1 megabyte per second

If necessary, it's possible to use light client

## Light Client Versus Full Node

- Light client does not store blocks or states, this way it needs less disk space (50 megabytes will be enough).
- Light client does not join p2p network and it does not produce any network cost when it is idle. The network overhead depends on how many requests the light client handles concurrently.
- Light client does not replay state of the chain so that there is not CPU cost when idle. The CPU cost also depends on how many requests the light client handles concurrently.
- Light client is faster than a full node even if it lagged behind the core network for a few months. It only needs a few seconds to catch up with core network.

2. Fortunately, there are some third party provider to access the BSC node. We can explore the transaction history and blocks on the chain, via bscscan, API and node RPC interfaces.

We can choose suitable service for our project

## 3rd Party Provider

- ANKR: <https://app.ankr.com/api>
- Chainstack: <https://chainstack.com/build-better-with-binance-smart-chain/>
- GetBlock.io: <https://getblock.io/nodes/bsc>
- QuikNode : <https://quiknode.io>

- Getblock

The screenshot displays the GetBlock website interface for Binance Smart Chain (BSC) nodes. The navigation bar includes links for Products, About, Pricing, Docs, SLA, Blog, and Contact. The main content area is titled 'Binance Smart Chain (BSC)' and features a 'PARTNER' badge. It lists endpoints for the mainnet, the current height (8780710), the location (Germany), and the connection speed (1 GB/sec). Below this, there are six pricing plans with 'Buy' buttons, and one 'Unlimited' plan with a 'Contact' button. A note at the bottom states that access to all available nodes is provided with a rate limit of 30 requests/sec.

Plan	Price	Requests	Action
500K	\$6	500K / Requests	Buy
1M	\$10	1M / Requests	Buy
5M	\$30	5M / Requests	Buy
10M	\$50	10M / Requests	Buy
50M	\$200	50M / Requests	Buy
Custom	Individual	Custom / Requests	Contact
Unlimited	\$500 / mo	Unlimited / Requests	Contact

Get access to all available nodes with a rate limit of 30 requests/sec.  
No limits are applied to the number of requests per day.

- Anyblock

**ANY**block

## Free Blockchain JSON-RPC Access for Binance Smart Chain (BSC)

fast, reliable & best value for money

- Fast & reliable BSC blockchain API
- Amazing support
- No need to run your own BSC node

Get started for free

```
-$ curl --location --request POST \  
> 'https://api.anyblock.tools/ethereum/ethereu\  
> --header 'Content-Type: application/json' \  
> --header 'Authorization: Bearer <your-token>  
> --data-raw '{  
>   "method": "eth_getBalance",  
>   "params": ["0xcCd9F664E49227976C7bf9Fcc7f"  
> }'
```

### ANYBLOCK API ACCESS BASIC

SUBSCRIBE	or	SUBSCRIBE
€199.00		€2,148.00
MONTHLY BILLING		annual billing (10% discount)

### ANYBLOCK API ACCESS PRO

SUBSCRIBE	or	SUBSCRIBE
€499.00		€5,388.00
MONTHLY BILLING		annual billing (10% discount)

- Quicknode

**QuickNode** Nodes Teams Support

#### LAUNCH

Monthly calls: **300 Thousand**  
Requests/sec: **No rate limiting**

Get your app live on the blockchain.  
Start with a 7-day free trial!

\$9.00 / mo **SELECT**

#### PRO

Monthly calls: **20 Million**  
Requests/sec: **No rate limiting**

Give your app more juice once it's live  
and growing.

\$99.00 / mo **SELECT**

#### SCALE

Monthly calls: **60 Million**  
Requests/sec: **No rate limiting**

The plan for Dapps starting to hit the  
growth curve.

\$299.00 / mo **SELECT**

#### DEDICATED

Monthly calls: **Unmetered**  
Requests/sec: **No rate limiting**

Customizable node with no daily  
limits, located anywhere you need it.

\$300.00 / mo **SELECT**

#### ENTERPRISE

Monthly calls: **Unlimited**  
Requests/sec: **Unlimited / second**

Tired of managing infrastructure  
yourself? We've got you!

Starting at \$2k+ /  
mo **CONTACT**

## Workflow

1. Get new token info

Sample code snip for listen pending transactions the latest block(web3.js)

```
1 // In case you are using Node.js
2 const Web3 = require('web3');
3 // Get Personal API
4 const API_KEY = '95cc2497-d50c-44c8-b574-4bd348811be8';
5 // Setting getblock node as HTTP provider
6 // const provider = new Web3.providers.HttpProvider("https://bsc.getblock.io/mainnet/?api_key=" + API_KEY);
7 // or as WebSocket provider
8 const provider = new Web3.providers.WebsocketProvider("wss://bsc.getblock.io/testnet/?api_key=" + API_KEY);
9 // Creating web3 instance with given provider
10 const web3 = new Web3(provider);
11 // Initializing web3.eth method
12 var block = web3.eth.getBlockNumber().then(console.log);
13 web3.eth.subscribe('pendingTransactions', function (error, result) {
14 // })
15 // })
16 // })
17 // })
18 // })
19 // })
```

Result:

```
E:\work\bsc_subscriber>node getblock.js
10221163
{
  blockHash: null,
  blockNumber: null,
  from: '0xc7a57270daAD312F2aB9086A4f3496f870790656',
  gas: 6700000,
  gasPrice: '10000000000',
  hash: '0x6780a4f2f38b0093482d5ddf61c85a1b39348eff62d6951f4404436117deb112',
  input: '0x51160630',
  nonce: 2801,
  to: '0x2B9A60061949dbc1D6FB6ccD38101Fb0EcAcCAC7',
  transactionIndex: null,
  value: '0',
  type: '0x0',
  v: '0xe5',
  r: '0x734cd4bcc329062e37fa9903ccfa0c6726243e51521135bdfa849c8c7e7ca550',
  s: '0x132ce37983d24170f02799ff38a9e0588bdf1a476387562cb7e6ce3fb62e0b83'
}
```

Get transaction receipt from tx hash

```
10 const web3 = new Web3(provider);
11 // Initializing web3.eth method
12 var block = web3.eth.getBlockNumber().then(console.log);
13 var receipt = web3.eth.getTransactionReceipt('0x6780a4f2f38b0093482d5ddf61c85a1b39348eff62d6951f4404436117deb112')
14 .then(console.log);
```

If contractAddress is not null, this is new contract and can access from address

[illegible]

Once got new address, we can get token details from bscscan or other API easily

Get Token Info by ContractAddress **PRO**

Only return token info for a token contract that is updated on BscScan

```
https://api.bscscan.com/api?module=token&action=tokeninfo&contractaddress=0xe09fab73bd3ade0a17ecc321fd13a19e81ce82&apikey=YourApiKeyToken
```

\* The above API endpoint is throttled to 2 calls/second regardless of API Pro tier.

Sample return of token info API:

```
{
  "status": "1",
  "message": "OK",
  "result": [
    {
      "contractAddress": "0x...",
      "tokenName": "Token Name",
      "symbol": "Token Symbol",
      "divisor": "18",
      "tokenType": "BEP20",
      "totalSupply": "10000000000000000",
      "blueCheckmark": "true",
      "description": "Token Description",
      "website": "https://token.website",
      "email": "email@token.website",
      "blog": "https://blog.token.website/",
      "reddit": "https://www.reddit.com/r/tokenwebsite/",
      "slack": "https://chat.token.website/"
    }
  ]
}
```

### Get Token Holder List by ContractAddress PRO

Return the current token holder and number of tokens held

```
https://api.bscscan.com/api?module=token&action=tokenholderlist&contractaddress=0x0e09fabb73bd3ade0a17ecc321fd13a19e81ce82&page=1&offset=10000&apikey=YourApikeyToken
```

Sample return of token info API:

```
{
  "status": "1",
  "message": "OK",
  "result": [
    {
      "TokenHolderAddress": "0x0000000000000000000000000000000000000000",
      "TokenHolderQuantity": "0"
    },
    {
      "TokenHolderAddress": "0x0000000000000000000000000000000000000001",
      "TokenHolderQuantity": "844975208812149014804348"
    },
    {
      "TokenHolderAddress": "0x0000000000000000000000000000000000000002",
      "TokenHolderQuantity": "120243030179598696424583296"
    }
  ]
}
```

## CoinMarketCap

### Technical Notes

- A historic quote for every "interval" period between your "time\_start" and "time\_end" will be returned.
- If a "time\_start" is not supplied, the "interval" will be applied in reverse from "time\_end".
- If "time\_end" is not supplied, it defaults to the current time.
- At each "interval" period, the historic quote that is closest in time to the requested time will be returned.
- If no historic quotes are available in a given "interval" period up until the next interval period, it will be skipped.

### Interval Options

There are 2 types of time interval formats that may be used for "interval".

The first are calendar year and time constants in UTC time:

- "hourly" - Get the first quote available at the beginning of each calendar hour.
- "daily" - Get the first quote available at the beginning of each calendar day.
- "weekly" - Get the first quote available at the beginning of each calendar week.
- "monthly" - Get the first quote available at the beginning of each calendar month.
- "yearly" - Get the first quote available at the beginning of each calendar year.

The second are relative time intervals.

"m": Get the first quote available every "m" minutes (60 second intervals). Supported minutes are: "5m", "10m", "15m", "30m", "45m".

```
200 Successful 400 Bad Request
401 Unauthorized 403 Forbidden
429 Too Many Requests 500 Internal Server Error

{
  "data": {
    "quotes": [
      {
        "timestamp": "2018-07-31T00:02:00Z",
        "btc_dominance": 47.9949,
        "active_cryptocurrencies": 2500,
        "active_exchanges": 600,
        "active_market_pairs": 1000,
        "quote": {
          "USD": {
            "total_market_cap": 292,
            "total_volume_24h": 176,
            "total_volume_24h_reported": 176,
            "altcoin_market_cap": 1,
            "altcoin_volume_24h": 3,
            "altcoin_volume_24h_reported": 3,
            "timestamp": "2018-07-31T00:02:00Z"
          }
        }
      }
    ]
  }
}
```

And I think it's necessary to get some items from several APIs

This is the sample code to integrate Metamask to website

```
function startProcess() {  
  if ($('#inp_amount').val() > 0) {  
    // run metamsk functions here  
    ETHAppDeploy.loadEthereum();  
  } else {  
    alert('Please Enter Valid Amount');  
  }  
}  
  
ETHAppDeploy = {  
  loadEthereum: async () => {  
    if (typeof window.ethereum !== 'undefined') {  
      ETHAppDeploy.web3Provider = ethereum;  
      ETHAppDeploy.requestAccount(ethereum);  
    } else {  
      alert(  
        "Not able to locate an Ethereum connection, please install a  
        Metamask wallet"  
      );  
    }  
  },  
  /**  
   * Request A Account  
   * **/  
  requestAccount: async (ethereum) => {  
    ethereum  
      .request({  
        method: 'eth_requestAccounts'  
      })  
  }  
}
```

Here You Must Address In: {

20

- [GitHub Repository](#)
- [Article In Dev.to](#)

Thank You VeryMuch




[Lathindu Pramuditha](#)

https://cdpn.io

### Connect With MetaMask

Select account(s)

Select all ⓘ New Account

<input type="checkbox"/>		Tester (...4ee5)	0 BNB
<input checked="" type="checkbox"/>		Randy (...d007)	0 BNB
<input type="checkbox"/>		Mekunka (...386f)	0 BNB

Only connect with sites you trust. [Learn more](#)

Cancel Next

I will research more suitable API and prepare all to start.

Please share me updated theme and charting view libraries.