

# Public docs

# Welcome

Welcome to TAAL Docs!!

You can use these docs to learn more about TAAL's core products, explore guides, and check our API references.

*You can get access to [TAAL PLATFORM](#) and get  PoweredByTAAL .*

## What is TAAL?

TAAL delivers value-added blockchain services, providing professional-grade, highly scalable blockchain infrastructure, and transactional solutions to support builders and businesses building on the BitcoinSV protocol.

## TAAL Services

### Transaction Processing

As a Transaction processor (or miner) TAAL validates clients transactions and time stamps them by sequencing and encoding them into blocks, which are then published and added to the BSV blockchain.

TAAL API is an easy to integrate, open communication channel to the blockchain. By connecting an application to this endpoint data, in the form of transactions, can be processed, instantly verified, and once settled this data can be viewed anytime. Creating great value to applications that require real time blockchain data at scale, now and into the future.

### Digital Asset Tokenization with the 1Sat Ordinals protocol

1SatOrdinals in an implementation of Ordinals running on the BSV blockchain. The tokens represent individual satoshis (the smallest unit of Bitcoin) that are "inscribed" with specific data, such as images or text. 1Sat Ordinals on BSV are fast, inexpensive, and fully scriptable via native Bitcoin Script.

1Sat Ordinals tokens API



## Other benefits

- The ability to have low-cost transactions (less than a penny), which make a difference when issuers are dropping millions of NFTs, for example. This attribute means more margin for the issuers on primary and secondary markets.
- Interoperability with ecosystem wallets and other blockchain platforms.
- Transparency allows the token to be tracked throughout its lifecycle, which makes a difference in a regulatory environment.
- No network congestion — as the tokens are issued at the network level on a blockchain that scales unbounded.

You can license this technology from TAAL, download the SDK, and start building their next game-changing NFT or digital asset platform today!



[Whitepaper](#)



[TAAL Website](#)



[Blog Post](#)



# Introduction

# Get an API Key

The following are the steps to get an API key from TAAL:

1. Enter the URL <https://platform.taal.com/> in your browser
2. Register or login on to **TAAL PLATFORM**.
3. Your mainnet and testnet API keys will be displayed.



# Core Products

# WhatsOnChain

This API accompanies the [WhatsOnChain.com](https://WhatsOnChain.com) website to form a suite of blockchain explorer services run on fully independent infrastructure.

## Overview

WhatsOnChain provides access to Bitcoin SV blocks, transactions, address activity, on-chain data, stats, insights, and much more via a simple REST API.

You can also use this API to broadcast transactions.

The supported networks are:

- Mainnet
- Testnet

 [Run in Postman](#)

## Authentication

At this stage no authentication is required to use the API up to 3 requests/sec. If you need more, see below on how to increase your rate limit using an API key.

## Rate Limits

Up to 3 requests/sec is free. Need more? Please create a free account on the TAAL Platform and view the [available subscription plans](#) there. We offer packages with 10, 20 or 40 requests/sec, and even a custom Enterprise plan.

If you are using the API for free, without an API key, please help us spread the word and use the "[Powered by WhatsOnChain](#)" logo in your app or website.

Once you get your API key from the Platform, here is a usage example:

```
curl -H 'Authorization: mainnet_xxxxxxxxxxxxxxxxxxxxxx' \  
https://api.whatsonchain.com/v1/bsv/main/chain/info
```

Please make sure to replace `mainnet_xxxxxxxxxxxxxxxxxxxxxx` with your actual API key.

If you need higher rate limits on `testnet`, please contact us through [Support](#).

## Support

Need help? Join our [WoC devs Telegram channel](#) and ask anything you need!

# Health

## Get API Status

A simple endpoint to show that the API server is up and running.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/woc
```

cURL

```
curl "https://api.whatsonchain.com/v1/bsv/main/woc"
```

The above request returns the below plain text response:

```
"Whats On Chain"
```

### URL Parameters

Parameter	Description
network	The selected network: main or test.

# Chain Info

## Get Blockchain Info

This endpoint retrieves information about the state of the chain for the selected network.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/chain/info
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/c
```

### Response JSON structure

```
{
  "chain": "main",
  "blocks": 635302,
  "headers": 635299,
  "bestblockhash": "000000000000000000002a40d7410a6c08109521c14f4cf354e7b3",
  "difficulty": 287310033717.7086,
  "mediantime": 1589703256,
  "verificationprogress": 0.9999754124031851,
  "pruned": false,
  "chainwork": "00000000000000000000000000000000000000000000000000000000000000010969f724913e0"
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Get Circulating Supply

This endpoint retrieves the circulating supply of BSV.

#### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/circulatingSupply
```

#### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/circulatingSupply"
```

#### Response

```
18227675
```

## URL Parameters

Parameter	Description
network	The selected network: main.

## Get Chain Tips

This endpoint retrieves information about all known tips in the block tree.

The possible values for status:

1. invalid: This branch contains at least one invalid block.
2. headers-only: Not all blocks for this branch are available, but the headers are valid.
3. valid-headers: All blocks are available for this branch, but they were never fully validated.
4. valid-fork: This branch is not part of the active chain, but is fully validated.
5. active: This is the tip of the active main chain, which is certainly valid.

#### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/chain/tips
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/c
```

## Response JSON structure

```
[  
  {  
    "height": 721622,  
    "hash": "000000000000000005693b8ef945a19fd269d2b19ebbad74e3d98d4427b2  
    "branchlen": 0,  
    "status": "active"  
  },  
  {  
    "height": 721213,  
    "hash": "00000000000000000478d6745b0f4060f1a00c143846a99b2dd6e1ee0e74  
    "branchlen": 1,  
    "status": "valid-fork"  
  },  
  ...  
]
```

## URL Parameters

network

The selected network: main.

# Get Peer Info

This endpoint retrieves information on peers connected to the node.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/peer/info
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/p
```

## Response JSON structure

```
[  
 {  
   "id": 326,  
   "addr": "<address>",  
   "addrlocal": "<address-local>",  
   "services": "0000000000000021",  
   "relaytxes": true,  
   "lastsend": 1665409946,  
   "lastrecv": 1665409976,  
   "bytessent": 890791261,  
   "bytesrecv": 2960980299,  
   "conntime": 1654039986,  
   "timeoffset": 0,  
   "pingtime": 0.000493,  
   "minping": 0.000303,  
   "version": 70016,  
   "subver": "/Bitcoin SV:1.0.11/",  
   "inbound": true,  
   "addnode": false,  
   "startingheight": 1493982,  
   "txninvsize": 0,  
   "banscore": 0,  
   "synced_headers": 1515937,  
   "synced_blocks": 1515937,  
   "whitelisted": false,  
   "bytessent_per_msg": {  
     "addr": 11220,  
     "blocktxn": 0,  
     "cmpctblock": 24060,  
     "feefilter": 0,  
     "getaddr": 0,  
     "getdata": 26657339,  
     "getheaders": 1085,  
     "headers": 876489,  
     "inv": 50080517,  
     "notfound": 61,  
     "ping": 3031616,  
     "pong": 3031648,  
     "reject": 664,  
     "sendcmpct": 66,  
     "sendheaders": 24,  
     "tx": 807054746,  
     "verack": 24,  
     "version": 147  
   },  
   "bytesrecv_per_msg": {  
     "addr": 55,  
     "blocktxn": 1174185732.  
   }  
 }
```

```
-----,
"cmpctblock": 1732490,
"feefilter": 192,
"getaddr": 24,
"getdata": 3864931,
"getheaders": 1085,
"headers": 1471198,
"inv": 133115380,
"notfound": 61,
"ping": 3031648,
"pong": 3031616,
"reject": 0,
"sendcmpct": 33,
"sendheaders": 24,
"tx": 471157544,
"verack": 24,
"version": 147
}
}
...
]
```

## URL Parameters

network

The selected network: main.

# Block

## Get by Hash

This endpoint retrieves the block details of a given hash.

- ⓘ For a block with up to 100 transactions, all transaction ids are returned in response to this call. If a block has more than 100 transactions, only the top 100 transaction ids are returned. To get remaining ids see the '[Get block pages](#)' section.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/block/hash/<hash>
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/b
```

Response JSON structure



```
        "asm": "OP_DUP OP_HASH160 85770ad0ee64a4f3ee4027e199c32f876e03a  
        "hex": "76a91485770ad0ee64a4f3ee4027e199c32f876e03a2d988ac",  
        "reqSigs": 1,  
        "type": 0,  
        "addresses": ["1DAhUiQxHzL1oAus0UDdUM89spXuAG1sqy"],  
        "opReturn": null  
    }  
}  
],  
"blockhash": "000000000000000000000004a288072ebb35e37233f419918f9783d499979  
"confirmations": 89,  
"time": 1553416668,  
"blocktime": 1553416668  
},  
"totalFees": 0.000238219999999557,  
"miner": "Bmgpool",  
"pages": null  
}
```

## URL Parameters

Parameter	Description
network	Selected network: main or test.
hash	The hash of the block to retrieve

## Get by Height

This endpoint retrieves the block details of a given block height.

- (i) For a block with up to 100 transactions, all transaction ids are returned in response to this call. If a block has more than 100 transactions, only the top 100 transaction ids are returned. To get remaining ids see the '[Get block pages](#)' section.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/block/height/<height>
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/b
```

## Response JSON structure



```
        "reqSigs": 1,
        "type": 0,
        "addresses": ["17fm4xevvwDh3XRHv9UoqYrVgPMbwGHSUs"],
        "opReturn": null
    },
}
],
"blockhash": "0000000000000000000091216c46973d82db057a6f9911352892b7769e",
"confirmations": 1,
"time": 1553501874,
"blocktime": 1553501874
},
"totalFees": 0.000004199999999343618,
"miner": "SVpool",
"pages": null
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
height	The height of the block to retrieve.

## Get Block Pages

If a block has more than 100 transactions, the page URLs will be provided in the pages element when getting a block by hash or height.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/block/hash/<hash>/page/
```

Example pages element in Get block by height or hash response:

```
{  
  "hash": "000000000000000000004a288072ebb35e37233f419918f9783d499979cb6ac  
  ...  
  "pages": {  
    "uri": [  
      "/block/hash/00000000000000000000885a4d8e9912f085b42288adc58b3e  
      "/block/hash/00000000000000000000885a4d8e9912f085b42288adc58b3e  
      ...  
    ],  
    "size": 50000  
  }  
}
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/csv/main/
```

## Response JSON structure

```
[  
  "89b4e01ea1ab27edb6ea66899637b526faebef42ca0302ff9ed916670f1330fe",  
  "1097bfc9b1f586352a1ab29af27c1f89e0a5d9790ea08a2b2a7d988ea6f2cfe",  
  ...  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The hash of the block to retrieve.
number	The page number.

## Get Header by Hash or Height

This endpoint retrieves the block header details of a given hash or height.

Use '`?format=block-headers-client`' as a query parameter for the [block-headers-client](#) response format.

Possible values for the status when the block-headers-client format is requested:

1. *active*: Block is a part of the current active chain.
2. *orphaned*: Block is not a part of the current active chain.

#### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/block/<hashOrHeight>/he
```

#### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/b
```

#### Response JSON structure

```
{
  "hash": "00000000000000004a288072ebb35e37233f419918f9783d499979cb6ac33",
  "confirmations": 68461,
  "height": 575045,
  "version": 536928256,
  "versionHex": "2000e000",
  "merkleroot": "4ebcba09add720991d03473f39dce4b9a72cc164e505cd446687a54",
  "time": 1553416668,
  "mediantime": 1553414858,
  "nonce": 87914848,
  "bits": "180997ee",
  "difficulty": 114608607557.4425,
  "chainwork": "0000000000000000000000000000000000000000000000000000000000ddff5d385546872b",
  "previousblockhash": "00000000000000000000988156c7075dc9147a5b62922f131086",
  "nextblockhash": "0000000000000000112b36a37c10235fa0c991f680bc5482ba96"
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The hash of the block to retrieve.

## Get Headers

This endpoint retrieves the last 10 block headers.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/block/headers
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/b
```

Response JSON structure

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# Get Header Bytes File Links

This endpoint retrieves a list of block header binary file links and each file only contains 80-byte block headers. These contain 10,000 block headers per file up to height 760,000.

Then 2,000 blocks per file after height 760,001. New files are automatically created after every 2,000 blocks and added to the list.

#### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/main/block/headers/resources
```

#### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/b
```

#### Response JSON structure

```
{"files": [  
    "https://api.whatsonchain.com/v1/bsv/main/block/headers/0_10000_header",  
    "https://api.whatsonchain.com/v1/bsv/main/block/headers/100001_110000_header",  
    "https://api.whatsonchain.com/v1/bsv/main/block/headers/110001_120000_header",  
    ...  
]
```

## Get Latest Header Bytes

This endpoint retrieves the latest specified number of block headers (up to 100) as a binary file. If "count" parameter is not provided, returns the latest header file, with up to 2000 block headers.

#### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/main/block/headers/latest?count=<
```

#### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/b
```

### URL Parameter

Parameter	Description
network	The selected network: main.
count (optional)	Number of headers requested, between 1 and 100.

# Transaction

## Get by Tx Hash

This endpoint retrieves the transaction details for a given transaction hash.

In the response body, if any output's hex size (*vout[x].scriptPubKey.hex*) exceeds 100KB, then the data for *vout[x].scriptPubKey.hex* and *vout[x].scriptPubKey.asm* is truncated and a flag *vout[x].scriptPubKey.isTruncated* is set to true.

For unconfirmed transactions, block and confirmations info won't be returned, as it doesn't exist.

A separate endpoint [Get Raw Transaction Output Data](#) can be used to fetch the full hex data, if required by the client application.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tx/hash/<hash>
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/
```

### Response JSON structure

```
{  
  "txid": "c1d32f28baa27a376ba977f6a8de6ce0a87041157cef0274b20bfda2b0d8df",  
  "hash": "c1d32f28baa27a376ba977f6a8de6ce0a87041157cef0274b20bfda2b0d8df",  
  "size": 113,  
  "version": 1,  
  "locktime": 0,  
  "vin": [  
    {  
      "txid": "",  
      "vout": 0,  
      "scriptSig": {  
        "asm": "... ",  
        "hex": "... "  
      },  
      "sequence": 4294967295,  
      "coinbase": "03d7c6082f7376706f6f6c2e636f6d2f3edff034600055b8467f00  
    }  
  ],  
  "vout": [  
    {  
      "value": 12.5000042,  
      "n": 0,  
      "scriptPubKey": {  
        "asm": "OP_DUP OP_HASH160 492558fb8ca71a3591316d095afc0f20ef7d42f",  
        "hex": "76a914492558fb8ca71a3591316d095afc0f20ef7d42f788ac", // (reqSigs": 1,  
        "type": 0,  
        "addresses": ["17fm4xevwDh3XRHv9UoqYrVgPMbwGHSUs"],  
        "opReturn": null,  
        "isTruncated": false  
      }  
    }  
  ],  
  "blockhash": "000000000000000091216c46973d82db057a6f9911352892b7769ed5",  
  "confirmations": 2,  
  "time": 1553501874,  
  "blocktime": 1553501874}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The hash/txId of the transaction to retrieve.

## Get Transaction Propagation Status

This endpoint returns the propagation status for a given transaction. It queries a random set of peers on the network and returns the number of peers that have the transaction in question.

This is an expensive call, so it's recommended only for critical or high-value transactions that might require checking the network propagation status. It is rate-limited to 10 requests per 10 seconds.

If the transaction has been found on zero peers, possible reasons may include:

- Propagation is still in progress;
- The transaction fee rate, or that of a preceding transaction, is relatively low, making it invisible to peers that are configured to accept only higher fee rates.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tx/hash/<hash>/propagat
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/<netw
```

### Response JSON structure

```
{  
  "queried_peers": 16,  
  "found_on_peers": 16  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The hash/txId of the transaction.

## Broadcast Transaction

You can broadcast a transaction using this endpoint. Returns the txId in response or an error message from the node, with header content-type: *text/plain*.

Keep in mind that this endpoint is meant for small-scale usage. For commercial or enterprise usage, please take a look at [ARC Endpoints \(beta\)](#), which are built for those types of use cases.

### HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/tx/raw
```

### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"txhex\": \"hex...\" }"
```

The JSON structure for the above post request:

```
{  
  "txhex": "hex..."  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Request Body

Parameter	Description
txhex	The raw transaction data in hex.

## Bulk Transaction Details

You can fetch details for multiple transactions in a single request.

- Max 20 transactions per request.

### HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/txs
```

### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"txids\" : [\"294cd1ebd5689fdee03509f92c32184c0f52f037d4046af"]}
```

The JSON structure for the above post request:

```
{  
  "txids": [  
    "294cd1ebd5689fdee03509f92c32184c0f52f037d4046af250229b97e0c8f1aa",  
    "91f68c2c598bc73812dd32d60ab67005eac498bef5f0c45b822b3c9468ba3258"  
  ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Bulk Transaction Status

You can get the status of multiple transactions in a single request.

- Max 20 transactions per request.

#### HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/txs/status
```

#### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"txids\" : [\"294cd1ebd5689fdee03509f92c32184c0f52f037d4046af250229b97e0c8f1aa", \
"91f68c2c598bc73812dd32d60ab67005eac498bef5f0c45b822b3c9468ba3258"]}
```

The JSON structure for the above post request:

```
{  
  "txids": [  
    "294cd1ebd5689fdee03509f92c32184c0f52f037d4046af250229b97e0c8f1aa",  
    "91f68c2c598bc73812dd32d60ab67005eac498bef5f0c45b822b3c9468ba3258"  
  ]  
}
```

#### Response JSON structure

```
[  
  {  
    "txid": "294cd1ebd5689fdee03509f92c32184c0f52f037d4046af250229b97e0c8",  
    "blockhash": "000000000000000004b5ce6670f2ff27354a1e87d0a01bf61f3307f",  
    "blockheight": 612251,  
    "blocktime": 1575841517,  
    "confirmations": 86479  
  },  
  {  
    "txid": "91f68c2c598bc73812dd32d60ab67005eac498bef5f0c45b822b3c9468ba",  
    "blockhash": "000000000000000002e8d4b4c0385abd195709c82f16d9917f081b7",  
    "blockheight": 617279,  
    "blocktime": 1578837295,  
    "confirmations": 81451  
  }]  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Decode Transaction

You can decode a raw transaction using this endpoint. Returns a json in response or an error message from the node.

### HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/tx/decode
```

### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"txhex\": \"hex...\" }"
```

### Post request body JSON structure

```
{  
  "txhex": "hex..."  
}
```

### Response JSON structure

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Request Body

Parameter	Description
txhex	The raw transaction data in hex.

## Download Receipt

You can download the transaction receipt (in PDF).

#### HTTP Request

```
GET https://<network>.whatsonchain.com/receipt/<hash>
```

#### cURL

```
curl --location --request GET "https://main.whatsonchain.com/receipt/4bd
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The hash/txId of the transaction.

## Get a Transaction as Binary

This endpoint returns the transaction as a binary for a given hash.

#### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tx/<hash>/bin
```

#### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/
```

#### Response (file)

```
c1d32f28baa27a376ba977f6a8de6ce0a87041157cef0274b20bfda2b0d8df96.bin
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The hash/txId of the transaction.

# Get Raw Transaction Data

This endpoint returns the raw hex of a transaction for a given hash.

## HTTP Request

GET <https://api.whatsonchain.com/v1/bsv/<network>/tx/<hash>/hex>

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/
```

Response with header content-type: text/plain:

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The hash/txId of the transaction.

# Bulk Raw Transaction Data

You can get the raw data of multiple transactions in hex in a single request.

- Max 20 transactions per request.

## HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/txs/hex
```

## cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"txids\" : [\"294cd1ebd5689fdee03509f92c32184c0f52f037d4046af250229b97e0c8f1aa", \
"91f68c2c598bc73812dd32d60ab67005eac498bef5f0c45b822b3c9468ba3258"]}
```

The JSON structure for the above post request:

```
{  
  "txids": [  
    "294cd1ebd5689fdee03509f92c32184c0f52f037d4046af250229b97e0c8f1aa",  
    "91f68c2c598bc73812dd32d60ab67005eac498bef5f0c45b822b3c9468ba3258"  
  ]  
}
```

Response JSON structure:

```
[  
  {  
    "txid": "294cd1ebd5689fdee03509f92c32184c0f52f037d4046af250229b97e0c8",  
    "hex": "0100000001...00000000",  
    "blockhash": "000000000000000000004b5ce6670f2ff27354a1e87d0a01bf61f3307f",  
    "blockheight": 612251,  
    "blocktime": 1575841517,  
    "confirmations": 86479  
  },  
  {  
    "txid": "91f68c2c598bc73812dd32d60ab67005eac498bef5f0c45b822b3c9468ba",  
    "hex": "0100000001...00000000",  
    "blockhash": "000000000000000000002e8d4b4c0385abd195709c82f16d9917f081b7",  
    "blockheight": 617279,  
    "blocktime": 1578837295,  
    "confirmations": 81451  
  }]  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# Get Raw Transaction Output Data

This endpoint returns raw hex of the transaction output for a given transaction ID and output index.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tx/<hash>/out/<index>/h
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/
```

Response with header content-type: text/plain:

```
76a914492558fb8ca71a3591316d095afc0f20ef7d42f788ac
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The hash/txId of the transaction.
index	The Output/Vout index.

# Bulk Raw Transaction Output Data

This endpoint returns raw hex of the transaction output for a list of up to 40 transaction IDs and up to 40 output indexes per transaction.

## HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/main/txs/vouts/hex
```

## cURL

```
curl --location --request POST 'https://api.whatsonchain.com/v1/bsv/main/' \
--header 'Content-Type: application/json' \
--data-raw '{
  "txids": [
    {
      "txid": "e8022f0904e636a7f4c1e561ca0af794b354f6fb71bdfb57b5ac",
      "vouts": [
        0,
        1
      ]
    }
  ]
}'
```

The JSON structure for the above post request:

```
{ "txids": [
  {
    "txid": "e8022f0904e636a7f4c1e561ca0af794b354f6fb71bdfb57b5ac",
    "vouts": [0,1]
  }
]
```

Response JSON structure:

```
[ {
  "txid": "e8022f0904e636a7f4c1e561ca0af794b354f6fb71bdfb57b5ac3528",
  "vout": {
    "0": {
      "scriptPubKey": {
        "hex": "76a9146b7f6a5d5677d1f3635e589b2eacc75d08dc6c4"
      }
    },
    "1": {
      "scriptPubKey": {
        "hex": "006a036478733575453364693965487a6e6b476668656"
      }
    }
  }
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Errors

Error	Description
"unknown" for that txID	If a given txID is unknown.
(Ignored)	If a given output is unknown.

# Get Merkle Proof

This endpoint returns the Merkle root of the branch for a confirmed transaction, in the [TSC](#) format. Returns *null* for unconfirmed transactions.

The old, non-TSC-format endpoint is no longer supported.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tx/<hash>/proof/tsc
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/
```

### Response JSON structure

```
[  
  {  
    "index": 0,  
    "txOrId": "c1d32f28baa27a376ba977f6a8de6ce0a87041157cef0274b20bfd  
    "target": "0000000000000000091216c46973d82db057a6f9911352892b7769  
    "nodes": [  
      "7e0ba1980522125f1f40d19a249ab3ae036001b991776813d25aebe08e8b  
      "1e3a5a8946e0caf07006f6c4f76773d7e474d4f240a276844f866bd09820  
    ]  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The hash/txId of the transaction.

# Mempool

## Get Mempool Info

This endpoint retrieves various information about the node's mempool, for the selected network.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/mempool/info
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/m
```

Response JSON structure

```
{  
  "bytes": 444832,  
  "maxmempool": 8196000000,  
  "mempoolminfee": 0,  
  "size": 142,  
  "usage": 647072  
}
```

### URL Parameters

Parameter	Description
network	The selected network: main or test.

## Get Mempool Transactions

This endpoint retrieves a list of txIds from the node's mempool, for the selected network.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/mempool/raw
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/mempool/raw"
```

## Response JSON structure

```
[  
    "0b0911bcb1f762634ed3ee171694d68049420c730296a7a592b29d46919c7739",  
    "195c95162a7f73f2ff9d4a5d8894f63ec44712b3b69fe1c3e53464048c6a39e3",  
    ...  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# (Un)Spent Transaction Outputs

These endpoints cover spent and unspent transaction outputs.

## Get Unspent UTXOs by Address

This endpoint retrieves a combined, ordered list of both confirmed and unconfirmed UTXOs for a given address. Returns up to 100k unconfirmed results in one request. Confirmed results are paginated if more than 1000 are available. The rest can be accessed using the provided next-page token.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/unsp
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

Response JSON structure

```
{  
  "address": "1L2F8wYxTRagCZLnsm2engg8ngGECSeuE5",  
  "script": "b3005d46af31c4b5675b73c17579b7bd366dfe10635b7b43ac111aea52  
  "result": [  
    {  
      "tx_pos": 1,  
      "tx_hash": "9c3801270d353066ed0553efb243bcc296d16e8b  
      "value": 98,  
      "isSpentInMempoolTx": false,  
      "hex": "76a914d0a69b53d9a1e229b51a60cc24814f8750b7446  
      "status": "unconfirmed"  
    },  
    ...  
    {  
      "height": 861936,  
      "tx_pos": 0,  
      "tx_hash": "46b080bf212f3b355cff7956db3adb53840ac550  
      "value": 100,  
      "isSpentInMempoolTx": true,  
      "status": "confirmed"  
    }  
  ],  
  "error": ""  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

# Get Unconfirmed UTXOs by Address

This endpoint retrieves an ordered list of unconfirmed UTXOs for a given address.  
Returns up to 100k results in one request.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/uncon
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

## Response JSON structure

```
{  
  "result": [  
    {  
      "tx_pos": 1,  
      "tx_hash": "93d8ae93a9e074c555d80e86c130b6cd4d5c7efa2795f91b2  
      "value": 181724,  
      "hex": "76a9148a32b77e8f9bce47179fe03b6894e3d37003099888ac"  
    }  
  ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

## Bulk Unconfirmed UTXOs by Address

This endpoint retrieves an ordered list of unconfirmed UTXOs for a given set of addresses - max 20 addresses per request. Returns up to 100 items in one request.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/addresses/unconfirmed/u
```

## cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/  
--header "Content-Type: application/json" \  
--data "{\"addresses\" : [\"16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP\", \"1KGH
```

## Request body JSON structure

```
{  
  "addresses":  
  [  
    "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP" ,  
    "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob"  
  ]  
}
```

Response JSON structure

```
[  
  {  
    "address": "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "unspent": [  
      {  
        "height": 657540,  
        "tx_pos": 1,  
        "tx_hash": "d75485c2329a533fd06b5f55a3f21644741c0258f2974  
        "value": 25000000  
      },  
      {  
        "height": 657542,  
        "tx_pos": 1,  
        "tx_hash": "55a656d50327ec3237fa6e821ab62294695cf508d631  
        "value": 25000000  
      }  
    ],  
    "error": ""  
  },  
  {  
    "address": "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob",  
    "unspent": [  
      {  
        "height": 658133,  
        "tx_pos": 1,  
        "tx_hash": "7ae43aac97396bc99616d8273c6cd9b57f017d6d49aca  
        "value": 25000000  
      },  
      {  
        "height": 658134,  
        "tx_pos": 1,  
        "tx_hash": "5b25a56bbb959f9cf4b3e48dbbe412bf5cc85e655d27f  
        "value": 25000000  
      }  
    ],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# Get Confirmed UTXOs by Address

This endpoint retrieves an ordered list of confirmed UTXOs for a given address. Pagination is available using the provided next-page token.

The " `isSpentInMempoolTx` " flag enables filtering of spent unconfirmed transactions from this set.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/confi
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

Response JSON structure

```
{  
  "result": [  
    {  
      "height": 797518,  
      "tx_pos": 0,  
      "tx_hash": "6cc9631ef3dad77eb0141134167f20469d0b4e61405de57fe",  
      "value": 181827,  
      "isSpentInMempoolTx": false  
    }  
  ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address
limit	Between 1 and 10000; default is 10000.
token	Provided next page token.

## Bulk Confirmed UTXOs by Address

This endpoint retrieves an ordered list of confirmed UTXOs for a given set of addresses - max 20 addresses per request. Returns up to 20 items in one request.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/addresses/confirmed/unspent
```

### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/confirmed/unspent" \
--header "Content-Type: application/json" \
--data "{\"addresses\" : [\"16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP\", \"1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob\"]}"
```

### Request body JSON structure

```
{  
  "addresses": [  
    "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob"  
  ]  
}
```

### Response JSON structure

```
[  
  {  
    "address": "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "unspent": [  
      {  
        "height": 657540,  
        "tx_pos": 1,  
        "tx_hash": "d75485c2329a533fd06b5f55a3f21644741c0258f2974  
        "value": 25000000  
      },  
      {  
        "height": 657542,  
        "tx_pos": 1,  
        "tx_hash": "55a656d50327ec3237fa6e821ab62294695cf508d631  
        "value": 25000000  
      }  
    ],  
    "error": ""  
  },  
  {  
    "address": "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob",  
    "unspent": [  
      {  
        "height": 658133,  
        "tx_pos": 1,  
        "tx_hash": "7ae43aac97396bc99616d8273c6cd9b57f017d6d49aca  
        "value": 25000000  
      },  
      {  
        "height": 658134,  
        "tx_pos": 1,  
        "tx_hash": "5b25a56bbb959f9cf4b3e48dbbe412bf5cc85e655d27f  
        "value": 25000000  
      }  
    ],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# (Deprecated) Get UTXOs by Address

This endpoint is powered by ElectrumX and is planned for deprecation in the near future.

This endpoint retrieves an ordered list of UTXOs for a given address.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/unsp
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

Returns a response as long as the response message is less than 1MB in size.

Response JSON structure

```
[  
  {  
    "height": 578325,  
    "tx_pos": 0,  
    "tx_hash": "62824e3af3d01113e9bce8b73576b833990d231357bd718385958  
    "value": 1250020815  
  },  
  {  
    "height": 578326,  
    "tx_pos": 0,  
    "tx_hash": "4b008a10eebcf70f384442e4e3147bc76c6e4f764b516b208e148  
    "value": 1251827826  
  },  
  ...  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

## (Deprecated) Bulk UTXOs by Address

**This endpoint is powered by ElectrumX and is planned for deprecation in the near future.**

This endpoint retrieves a list of UTXOs for multiple addresses in a single request.

- Max 20 addresses per request.

### HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/addresses/unspent
```

### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"addresses\" : [\"16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP\", \"1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob"]}
```

The JSON structure for the above post request:

### JSON structure

```
{  
  "addresses": [  
    "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob"  
  ]  
}
```

### Response JSON structure

```
[  
  {  
    "address": "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "unspent": [  
      {  
        "height": 657540,  
        "tx_pos": 1,  
        "tx_hash": "d75485c2329a533fd06b5f55a3f21644741c0258f2974  
        "value": 25000000  
      },  
      {  
        "height": 657542,  
        "tx_pos": 1,  
        "tx_hash": "55a656d50327ec3237fa6e821ab62294695cf508d631  
        "value": 25000000  
      }  
    ],  
    "error": ""  
  },  
  {  
    "address": "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob",  
    "unspent": [  
      {  
        "height": 658133,  
        "tx_pos": 1,  
        "tx_hash": "7ae43aac97396bc99616d8273c6cd9b57f017d6d49aca  
        "value": 25000000  
      },  
      {  
        "height": 658134,  
        "tx_pos": 1,  
        "tx_hash": "5b25a56bbb959f9cf4b3e48dbbe412bf5cc85e655d27f  
        "value": 25000000  
      }  
    ],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# Get Unspent UTXOs by Script

This endpoint retrieves a combined, ordered list of both confirmed and unconfirmed UTXOs for a given script. Returns up to 100k unconfirmed results in one request. Confirmed results are paginated if more than 1000 are available. The rest can be accessed using the provided next-page token.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/script/<scriptHash>/uns
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/s
```

## Response JSON structure

```
{
  "script": "b3005d46af31c4b5675b73c17579b7bd366dfe10635b7b43ac111aea52",
  "result": [
    {
      "tx_pos": 1,
      "tx_hash": "9c3801270d353066ed0553efb243bcc296d16e8b",
      "value": 98,
      "isSpentInMempoolTx": false,
      "hex": "76a914d0a69b53d9a1e229b51a60cc24814f8750b7446",
      "status": "unconfirmed"
    },
    ...
    {
      "height": 861936,
      "tx_pos": 0,
      "tx_hash": "46b080bf212f3b355cff7956db3adb53840ac550",
      "value": 100,
      "isSpentInMempoolTx": true,
      "status": "confirmed"
    }
  ],
  "error": ""
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
scriptHash	The script hash: <i>Sha256</i> hash of the binary bytes of the locking script ( <i>ScriptPubKey</i> ), expressed as a hexadecimal string.

## Get Unconfirmed UTXOs by Script

This endpoint retrieves the ordered list of unconfirmed UTXOs for a given script. Returns up to 100k results in one request.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/script/<scriptHash>/unc
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/s
```

### Response JSON structure

```
{
  "result": [
    {
      "tx_pos": 1,
      "tx_hash": "93d8ae93a9e074c555d80e86c130b6cd4d5c7efa2795f91b2",
      "value": 181724,
      "hex": "76a9148a32b77e8f9bce47179fe03b6894e3d37003099888ac"
    }
  ]
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
scriptHash	The script hash: <i>Sha256</i> hash of the binary bytes of the locking script ( <i>ScriptPubKey</i> ), expressed as a hexadecimal string.

## Bulk Unconfirmed UTXOs by Script

This endpoint retrieves an ordered list of unconfirmed UTXOs for a given set of scripthashes - max 20 scripthashes per request. Returns up to 20 items in one request.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/scripts/unconfirmed/uns
```

### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"scripts\" : [\"f814a7c3a40164aacc440871e8b7b14eb6a45f0ca7dcb
```

### Request body JSON structure

```
{  
  "scripts": [  
    "f814a7c3a40164aacc440871e8b7b14eb6a45f0ca7dcbeaea709edc83274c5e7  
    "995ea8d0f752f41cdd99bb9d54cb004709e04c7dc4088bcbbbb9ea5c390a43c3  
  ]  
}
```

### Response JSON structure

```
[  
  {  
    "script": "f814a7c3a40164aacc440871e8b7b14eb6a45f0ca7dcbeaea709ed",  
    "unspent": [  
      {  
        "height": 620539,  
        "tx_pos": 0,  
        "tx_hash": "4ec3b63d764558303eda720e8e51f69bbcf813760756",  
        "value": 450000  
      }  
    ],  
    "error": ""  
  },  
  {  
    "script": "995ea8d0f752f41cdd99bb9d54cb004709e04c7dc4088bcbbbb9ea",  
    "unspent": [],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# Get Confirmed UTXOs by Script

This endpoint retrieves the ordered list of confirmed UTXOs for a given script. Pagination is available using the provided next-page token.

The "`isSpentInMempoolTx`" flag enables filtering of spent unconfirmed transactions from this set.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/script/<scriptHash>/con
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/s
```

Response JSON structure

```
{  
  "result": [  
    {  
      "height": 797518,  
      "tx_pos": 0,  
      "tx_hash": "6cc9631ef3dad77eb0141134167f20469d0b4e61405de57fe",  
      "value": 181827,  
      "isSpentInMempoolTx": false  
    }  
  ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
scriptHash	The script hash: <i>Sha256</i> hash of the binary bytes of the locking script ( <i>ScriptPubKey</i> ), expressed as a hexadecimal string.
limit	Between 1 and 10000; default is 10000.
token	Provided next page token.

# Bulk Confirmed UTXOs by Script

This endpoint retrieves an ordered list of confirmed UTXOs for a given set of scripthashes - max 20 scripthashes per request. Returns up to 20 items in one request.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/scripts/confirmed/unsp
```

## cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/"  
--header "Content-Type: application/json" \  
--data "{\"scripts\" : [\"f814a7c3a40164aacc440871e8b7b14eb6a45f0ca7dcb
```

## Request body JSON structure

```
{  
  "scripts": [  
    "f814a7c3a40164aacc440871e8b7b14eb6a45f0ca7dcbeaea709edc83274c5e7  
    "995ea8d0f752f41cdd99bb9d54cb004709e04c7dc4088bcbbbb9ea5c390a43c3  
  ]  
}
```

## Response JSON structure

```
[  
  {  
    "script": "f814a7c3a40164aacc440871e8b7b14eb6a45f0ca7dcbeaea709ed  
    "unspent": [  
      {  
        "height": 620539,  
        "tx_pos": 0,  
        "tx_hash": "4ec3b63d764558303eda720e8e51f69bbcfe813760756  
        "value": 450000  
      }  
    ],  
    "error": ""  
  },  
  {  
    "script": "995ea8d0f752f41cdd99bb9d54cb004709e04c7dc4088bcbbbb9ea  
    "unspent": [],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# (Deprecated) Get UTXOs by Script

This endpoint is powered by ElectrumX and is planned for deprecation in the near future.

This endpoint retrieves the ordered list of UTXOs for a given script.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/script/<scriptHash>/un
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/s
```

Returns a response as long as the response message is less than 1MB in size.

Response JSON structure

```
[  
  {  
    "height": 578325,  
    "tx_pos": 0,  
    "tx_hash": "62824e3af3d01113e9bce8b73576b833990d231357bd718385958  
    "value": 1250020815  
  },  
  {  
    "height": 578326,  
    "tx_pos": 0,  
    "tx_hash": "4b008a10eebcf70f384442e4e3147bc76c6e4f764b516b208e148  
    "value": 1251827826  
  },  
  ...  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
scriptHash	The script hash: <i>Sha256</i> hash of the binary bytes of the locking script ( <i>ScriptPubKey</i> ), expressed as a hexadecimal string.

## (Deprecated) Bulk UTXOs by Script

This endpoint is powered by ElectrumX and is planned for deprecation in the near future.

This endpoint retrieves a list of UTXOs for multiple scripts in a single request.

- Max 20 scripts per request.

HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/scripts/unspent
```

cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"scripts\": [\"f814a7c3a40164aacc440871e8b7b14eb6a45f0ca7dcbeaea709edc83274c5e7",
"995ea8d0f752f41cdd99bb9d54cb004709e04c7dc4088bcbbbb9ea5c390a43c3"]}
```

The JSON structure for the above post request:

JSON structure

```
{  
  "scripts": [  
    "f814a7c3a40164aacc440871e8b7b14eb6a45f0ca7dcbeaea709edc83274c5e7",  
    "995ea8d0f752f41cdd99bb9d54cb004709e04c7dc4088bcbbbb9ea5c390a43c3"  
  ]  
}
```

Response JSON structure

```
[  
  {  
    "script": "f814a7c3a40164aacc440871e8b7b14eb6a45f0ca7dcbeaea709ed",  
    "unspent": [  
      {  
        "height": 620539,  
        "tx_pos": 0,  
        "tx_hash": "4ec3b63d764558303eda720e8e51f69bbcf813760756",  
        "value": 450000  
      }  
    ],  
    "error": ""  
  },  
  {  
    "script": "995ea8d0f752f41cdd99bb9d54cb004709e04c7dc4088bcbbbb9ea",  
    "unspent": [],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# Get Unconfirmed Spent Tx Output

This endpoint retrieves where the specified unconfirmed transaction output was spent.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tx/<txid>/<voutIndex>/u
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

### Response JSON structure

```
{  
    "txid": "b905857a36586e68b8186be1e7b1ac56de196f5257536f019d7071cbdb06  
    "vin": 0  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
txid	The transaction ID.
voutIndex	The specific output index.

# Get Confirmed Spent Tx Output

This endpoint retrieves where the specified confirmed transaction output was spent.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tx/<txid>/<voutIndex>/c
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

### Response JSON structure

```
{  
    "txid": "87645a7c4a9a5a9b9cd1468db19cdba44129c35cc487fc57c4d8843141ad  
    "vin": 2  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
txid	The transaction ID.
voutIndex	The specific output index.

## Get Spent Transaction Output

This endpoint retrieves where the specified transaction output was spent, checking both confirmed and unconfirmed in a single call.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tx/<txid>/<voutIndex>/s
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

### Response JSON structure

```
{  
  "txid": "87645a7c4a9a5a9b9cd1468db19cdba44129c35cc487fc57c4d8843141ad  
  "vin": 2  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
txid	The transaction ID.
voutIndex	The specific output index.

## Errors

Error	Description
400 Bad Request	If UTXO is unknown.
404 Not Found	If UTXO is known but spent details are not found, i.e., it's still unspent.

## Bulk Spent Transaction Outputs

This endpoint retrieves where the specified transaction outputs were spent, checking both confirmed and unconfirmed in a single call.

- Max 20 transaction outputs per request.

### HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/utxos/spent
```

### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"utxos\": [{\"txid\": \"87645a7c4a9a5a9b9cd1468db19cdba44129",
```

### Request body JSON structure

```
{
  "utxos": [
    {
      "txid": "87645a7c4a9a5a9b9cd1468db19cdba44129c35cc487fc57c4d88431",
      "vout": 2
    },
    {
      "txid": "e05de19075bc8205d702036a53d3e21ff6a2964f1e9941b913921e1e",
      "vout": 0
    }
  ]
}
```

### Response JSON structure

```
[  
  {  
    "utxo": {  
      "txid": "87645a7c4a9a5a9b9cd1468db19cdba44129c35cc487fc57c4d8  
      "vout": 2  
    },  
    "spentIn": {  
      "txid": "88d8ce011179fac0dc466a1c5fcb734a91b076d15354aee4f718  
      "vin": 93,  
      "status": "confirmed"  
    },  
    "error": ""  
  },  
  {  
    "utxo": {  
      "txid": "e05de19075bc8205d702036a53d3e21ff6a2964f1e9941b91392  
      "vout": 0  
    },  
    "spentIn": {  
      "txid": "81f4ffcdc8ac15a9f360196f95f8e4e32c2e2c4fb78fdca64e52  
      "vin": 0,  
      "status": "unconfirmed"  
    },  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Errors

Parameter	Description
400 Bad Request	If UTXO is unknown.
404 Not Found	If UTXO is known but spent details are not found, i.e., it's still unspent.

# Address

## Get Address Info

This endpoint retrieves various information for a given address.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/info
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

Response JSON structure

```
{  
    "address": "16ZqP5Tb22KJuVSAbjNkoIZs13mmRmexZA",  
    "ismine": false,  
    "isscript": false,  
    "isValid": true,  
    "iswatchonly": false,  
    "scriptPubKey": "76a9143d0e5368bdaddca108a0fe44739919274c726c788ac"  
}
```

### URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

## Get Address Usage Status

This endpoint serves as a usage status flag for a given address.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/used
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

## Response JSON structure

```
true
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

## Get Associated Scripthashes

There are some addresses that are associated with multiple script types. This endpoint returns a list of scripthashes associated to the given address, and their types.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/script
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

## Response JSON structure

```
[  
  {  
    "script": "55f4cd3dabddd62707b374f69f66b97f5f53dac5b204bd6c09dcad  
    "type": "pubkeyhash"  
  },  
  {  
    "script": "cc0d7111befe9f24f1824c797e837c5cff2f8773901ccb9aca0372  
    "type": "pubkey"  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

# Download Statement

You can download the statement (in PDF) for a given address.

### HTTP Request

```
GET https://<network>.whatsonchain.com/statement/<address>
```

### cURL

```
curl --location --request GET "https://main.whatsonchain.com/statement/1"
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

# Get Unconfirmed Balance

This endpoint retrieves the unconfirmed balance for a given address.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/uncon
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

## Response JSON structure

```
{  
    "unconfirmed": 103  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

# Bulk Unconfirmed Balance

This endpoint retrieves the unconfirmed balance for multiple addresses in a single request.

- Max 20 addresses per request.

## HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/addresses/unconfirmed/
```

## cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/"  
--header "Content-Type: application/json" \  
--data "{\"addresses\" : [\"16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP\"] }"
```

## Request body JSON structure

```
{  
  "addresses":  
  [  
    "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP"  
  ]  
}
```

## Response JSON structure

```
[  
  {  
    "address": "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "unconfirmed": 0,  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# Get Confirmed Balance

This endpoint retrieves the confirmed balance for a given address.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/confi
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

Response JSON structure

```
{  
    "confirmed": 181827  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

## Bulk Confirmed Balance

This endpoint retrieves the confirmed balance for multiple addresses in a single request.

- Max 20 addresses per request.

HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/addresses/confirmed/ba
```

cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/"  
--header "Content-Type: application/json" \  
--data "{\"addresses\": [\"1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob\", \"1ApL
```

Request body JSON structure

```
{  
  "addresses":  
  [  
    "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob",  
    "1ApLMk225o7S9FvKwpNChB7CX8cknQT9Hy"  
  ]  
}
```

Response JSON structure

```
[  
  {  
    "address": "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob",  
    "script": "4f8f588fc9bd7304dc73a14c5d3d0813e048986465e22587e3166c",  
    "confirmed": 0,  
    "error": ""  
  },  
  {  
    "address": "1ApLMk225o7S9FvKwpNChB7CX8cknQT9Hy",  
    "script": "9b15c6471cb6f0314250e9ff180b4197a9c409c2791261ebe4e0c4",  
    "confirmed": 688992295481,  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# (Deprecated) Get Balance

This endpoint is powered by ElectrumX and is planned for deprecation in the near future.

This endpoint retrieves both the confirmed and unconfirmed balance for a given address.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/balan
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

Response JSON structure

```
{  
    "confirmed":533134093647,  
    "unconfirmed":0  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

## (Deprecated) Bulk Balance

This endpoint is powered by ElectrumX and is planned for deprecation in the near future.

This endpoint retrieves both the confirmed and unconfirmed balance for multiple addresses in a single request.

- Max 20 addresses per request.

HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/addresses/balance
```

cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/"  
--header "Content-Type: application/json" \  
--data "{\"addresses\" : [\"16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP\", \"1KGH
```

The JSON structure for the above post request:

JSON structure

```
{  
  "addresses": [  
    "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob"  
  ]  
}
```

Response JSON structure

```
[  
  {  
    "address": "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "balance": {  
      "confirmed": 450000,  
      "unconfirmed": 0  
    },  
    "error": ""  
  },  
  {  
    "address": "1KGHhLTQaPr4LErrvbAuGE62yPpDoRwrob",  
    "balance": {  
      "confirmed": 0,  
      "unconfirmed": 0  
    },  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# Get Unconfirmed History

This endpoint retrieves unconfirmed transactions for a given address. Returns up to 100k results in one request.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/uncon
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

Response JSON structure

```
{  
    "result": [  
        {  
            "tx_hash": "93d8ae93a9e074c555d80e86c130b6cd4d5c7efa2795f91b2"  
        },  
        {  
            "tx_hash": "6bcd7708adf5c8c00d299628ef86fa5d982ccd1909254f46e"  
        },  
        {  
            "tx_hash": "cf1f1fd7962aeac56932b0e522d83c163d89873f2e26a1519"  
        },  
        ...  
    ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

# Bulk Unconfirmed History

This endpoint retrieves the history of unconfirmed transactions for a given set of addresses.

- Max 20 addresses per request.
- Max 100 items returned per request.

#### HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/addresses/unconfirmed/
```

#### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"addresses\" : [ \"16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP\" ] }"
```

#### Request body JSON structure

```
{  
  "addresses": [  
    "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP"  
  ]  
}
```

#### Response JSON structure

```
[  
  {  
    "address": "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "script": "c7713679ea48e31aec663612f5cbaeb86f9241677c0397b42d9575  
    "result": [  
      {  
        "tx_hash": "6216506e2b5ef029595e1b29a20c279889873a9da4e98  
        "height": 658093  
      },  
      {  
        "tx_hash": "df860638d1530d2ba1f25d167fa24bbef3b1387e94207  
        "height": 658093  
      }  
    ],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Get Confirmed History

This endpoint retrieves confirmed transactions for a given address. Pagination is available using the provided next-page token.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/confi
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

### Response JSON structure

```
{  
  "result": [  
    {  
      "tx_hash": "6cc9631ef3dad77eb0141134167f20469d0b4e61405de57fe",  
      "height": 797518  
    }  
  ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address
order	Ordering: asc or desc; default is desc.
limit	Between 1 and 1000; default is 100.
height	Starting block height for history; default is 0.
token	Provided next page token.

## Bulk Confirmed History

This endpoint retrieves the history of confirmed transactions for a given set of addresses.

- Max 20 addresses per request.
- Max 20 items returned per request.

For pagination please use the single address endpoint.

### HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/addresses/confirmed/hi
```

### cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"addresses\" : [ \"16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP\" ] }"
```

### Request body JSON structure

```
{  
  "addresses": [  
    "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP"  
  ]  
}
```

Response JSON structure

```
[  
  {  
    "address": "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "script": "c7713679ea48e31aec663612f5cbaeb86f9241677c0397b42d9575  
    "result": [  
      {  
        "tx_hash": "6216506e2b5ef029595e1b29a20c279889873a9da4e98  
        "height": 658093  
      },  
      {  
        "tx_hash": "df860638d1530d2ba1f25d167fa24bbef3b1387e94207  
        "height": 658093  
      }  
    ],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Bulk History

This endpoint retrieves the history of both confirmed and unconfirmed transactions for a given set of addresses.

- Max 20 addresses per request.
- Max 1000 confirmed and max 1000 unconfirmed history items returned per request.

- Page tokens provided if address contains more items than above.
- Call standard individual [unconfirmed](#) and [confirmed](#) endpoints with page token for more history if required.

## HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/addresses/history/all
```

## cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"addresses\" : [ \"16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP\" , \"19SsDUJ29XM2VDCkZKnTXJ9Zjt3gMtnrwC\" ]}
```

## Request body JSON structure

```
{  
    "addresses": [  
        "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP" ,  
        "19SsDUJ29XM2VDCkZKnTXJ9Zjt3gMtnrwC"  
    ]  
}
```

## Response JSON structure

```
[  
  {  
    "address": "16ZBEb7pp6mx5EAGrdeKivztd5eRJFuvYP",  
    "script": "c7713679ea48e31aec663612f5cbaeb86f9241677c0397b42d95754  
    "unconfirmed":  
      {  
        "result": [],  
        "error": ""  
      },  
    "confirmed":  
      {  
        "result": [  
          {  
            "tx_hash": "6216506e2b5ef029595e1b29a20c279889873a",  
            "height": 658093  
          },  
          {  
            "tx_hash": "df860638d1530d2ba1f25d167fa24bbef3b138",  
            "height": 658093  
          }  
        ],  
        "error": ""  
      }  
  },  
  {  
    "address": "19SsDUJ29XM2VDCKnTXJ9Zjt3gMtnrwC",  
    "script": "9ad0daaac33264199ce40596970c5ccf0096eb05b9b020002a3f1c2  
    "unconfirmed":  
      {  
        "result": [],  
        "error": ""  
      },  
    "confirmed":  
      {  
        "result": [  
          {  
            "tx_hash": "06f978c8758b45aa298e74d2951123811d",  
            "height": 825487  
          },  
          {  
            "tx_hash": "064cad8a9ff3caad45579ee514d6a352f4",  
            "height": 825488  
          }  
        ],  
        "error": ""  
      }  
  }]
```

]

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## (Deprecated) Get History

**This endpoint is powered by ElectrumX and is planned for deprecation in the near future.**

This endpoint retrieves both confirmed and unconfirmed transactions for a given address.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/histo
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

Returns a response as long as the response message is less than 1MB in size.

### Response JSON structure

```
[  
  {  
    "tx_hash": "2b5ccedd76ba0d524d95ecce1c8a76e5888e0f13591734be0169b  
    "height": 573216  
  },  
  {  
    "tx_hash": "7fe815e14b32aaaecc2cca7ed605ed612b532765710917cb711ea  
    "height": 573229  
  },  
  {  
    "tx_hash": "b28dd1723d0d1fa6e4b2875f7419028c1c1441fc02469aebc444c  
    "height": 573230  
  },  
  ...  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
address	The address

# Script

## Get Script Usage Status

This endpoint serves as a usage status flag for a given script.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/script/<scriptHash>/use
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/s
```

### Response JSON structure

```
false
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
scriptHash	The script hash: <i>Sha256</i> hash of the binary bytes of the locking script ( <i>ScriptPubKey</i> ), expressed as a hexadecimal string.

## Get Unconfirmed Script History

This endpoint retrieves the history of unconfirmed transactions for a given script.  
Returns up to 100k results in one request.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/script/<scriptHash>/unc
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/s
```

## Response JSON structure

```
{  
  "result": [  
    {  
      "tx_hash": "93d8ae93a9e074c555d80e86c130b6cd4d5c7efa2795f91b2"  
    },  
    {  
      "tx_hash": "6bcd7708adf5c8c00d299628ef86fa5d982ccd1909254f46e"  
    },  
    {  
      "tx_hash": "cf1f1fd7962aeac56932b0e522d83c163d89873f2e26a1519"  
    },  
    ...  
  ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
scriptHash	The script hash: <i>Sha256</i> hash of the binary bytes of the locking script ( <i>ScriptPubKey</i> ), expressed as a hexadecimal string.

# Bulk Unconfirmed Script History

This endpoint retrieves the history of unconfirmed transactions for a given set of scripts.

- Max 20 scripthashes per request.
- Max 100 items returned per request.

## HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/scripts/unconfirmed/hi
```

## cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/" \
--header "Content-Type: application/json" \
--data "{\"scripts\" : [ \"8ca0d228066cfffc2b7f31c7db7737185be9d06ff5c07206b61f06d80a4cdd9a8"]}
```

## Request body JSON structure

```
{  
  "scripts": [  
    "8ca0d228066cfffc2b7f31c7db7737185be9d06ff5c07206b61f06d80a4cdd9a8  
  ]  
}
```

## Response JSON structure

```
[  
  {  
    "script": "8ca0d228066cfffc2b7f31c7db7737185be9d06ff5c07206b61f06d  
    "result": [  
      {  
        "tx_hash": "4e6be25adf9b2bdf90a9f88e966f7cb5e9d71a22325c8  
      }  
    ],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

## Get Confirmed Script History

This endpoint retrieves the history of confirmed transactions for a given script. Pagination is available using the provided next-page token.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/script/<scriptHash>/con
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/s
```

## Response JSON structure

```
{  
    "result": [  
        {  
            "tx_hash": "6cc9631ef3dad77eb0141134167f20469d0b4e61405de57fe",  
            "height": 797518  
        }  
    ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
scriptHash	The script hash: <i>Sha256</i> hash of the binary bytes of the locking script ( <i>ScriptPubKey</i> ), expressed as a hexadecimal string.
order	Ordering: asc or desc; default is desc.
limit	Between 1 and 1000; default is 100.
height	Starting block height for history; default is 0.
token	Provided next page token.

## Bulk Confirmed Script History

This endpoint retrieves the history of confirmed transactions for a given set of scripts.

- Max 20 scripthashes per request.
- Max 20 items returned per request.

## HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/scripts/confirmed/hist
```

## cURL

```
curl --location --request POST "https://api.whatsonchain.com/v1/bsv/main/"  
--header "Content-Type: application/json" \  
--data "{\"scripts\" : [ \"8ca0d228066cfffc2b7f31c7db7737185be9d06ff5c07
```

## Request body JSON structure

```
{  
  "scripts": [  
    "8ca0d228066cfffc2b7f31c7db7737185be9d06ff5c07206b61f06d80a4cdd9a8  
  ]  
}
```

## Response JSON structure

```
[  
  {  
    "script": "8ca0d228066cfffc2b7f31c7db7737185be9d06ff5c07206b61f06d",  
    "result": [  
      {  
        "tx_hash": "4e6be25adf9b2bdf90a9f88e966f7cb5e9d71a22325c8"  
      }  
    ],  
    "error": ""  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

# (Deprecated) Get Script History

This endpoint is powered by ElectrumX and is planned for deprecation in the near future.

This endpoint retrieves both the confirmed and unconfirmed transactions for a given script.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/script/<scriptHash>/his
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/s
```

Returns a response as long as the response message is less than 1MB in size.

Response JSON structure

```
[  
  {  
    "tx_hash": "52dfceb815ad129a0fd946e3d665f44fa61f068135b9f38b05d3c  
    "height": 620539  
  },  
  {  
    "tx_hash": "4ec3b63d764558303eda720e8e51f69bbcf81376075657313fb5  
    "height": 620539  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
scriptHash	The script hash: <i>Sha256</i> hash of the binary bytes of the locking script ( <i>ScriptPubKey</i> ), expressed as a hexadecimal string.

# Exchange Rate

## Get Exchange Rate

This endpoint provides the current exchange rate for BSV.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/exchangerate
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/e
```

### Response

```
{  
  "currency": "USD",  
  "rate": "229.3163333333335",  
  "time": 1660176000  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main.

## Get Historical Exchange Rate

This endpoint provides the historical exchange rate data for BSV. Exchange rate data goes back to *2018/11/19*.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/main/exchangerate/historical?from
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/e
```

## Response

```
[  
  {  
    "rate": 60.3905,  
    "time": 1660089600  
  },  
  {  
    "rate": 62.661,  
    "time": 1660176000  
  },  
  ...  
]
```

## URL Parameters

Parameter	Description
from	The starting date ( <i>unixtimestamp</i> ) to retrieve the exchange rate data.
to	The end date ( <i>unixtimestamp</i> ) to retrieve the exchange rate data.

# Search

## Get Explorer Links

This endpoint identifies whether the posted query text is a block hash, txid, or address, and responds with WoC links. Ideal for extending customized search in apps.

### HTTP Request

```
POST https://api.whatsonchain.com/v1/bsv/<network>/search/links
```

### cURL

```
curl -X POST https://api.whatsonchain.com/v1/bsv/main/search/links -H '
```

### Response JSON structure

```
{
  "results": [
    {
      "type": "address",
      "url": "https://whatsonchain.com/address/1GJ3x5bcEnKMnzNFPPEL"
    }
  ]
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.

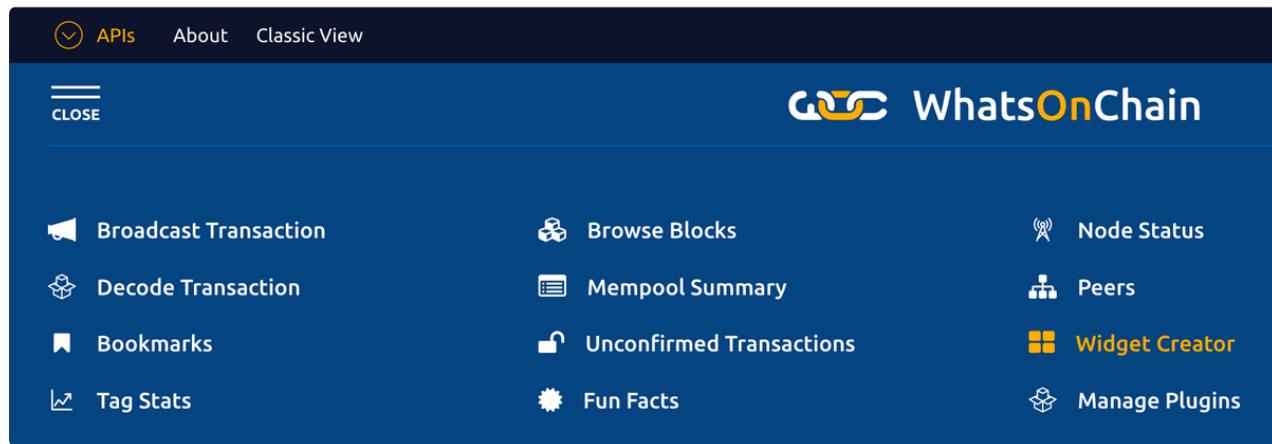
## Request Body

Parameter	Description
query	The text to search.

# WoC Widgets

You can embed aspects of the WhatsOnChain Home, Transaction, or Block pages in any application.

Go to the URL - [WhatsOnChain.com](https://WhatsOnChain.com), Click the **TOOLS** menu, and there you can find the **Widget Creator**.



You can use the **Widget Creator** tool to toggle sections of the selected page and can immediately see what it will look like in the preview section.

Please select page  
Home

Please select a theme  
Light

Hide Header  
 Hide Footer  
 Hide Search  
 Hide Latest Blocks  
 Hide Summary  
 Hide Block list  
 Hide Stats Summary  
 Hide Nav Tools

Url  
`https://whatsonchain.com/?theme=light&hideHeader=true&hideFooter=true&hideSearch=true&hideLatestBlocks=true&hideBlockList=true&hideStatsSummary=true&hideNavTools=true&hideTxSummary=true`

**Mainnet Summary**

<b>Hashrate</b> 442.639 PH/s	<b>Unconfirmed Transactions</b> 621 tx Estimated block size: 1.68 MB Mempool Usage: 15.54 MB	<b>Difficulty</b> 59.546 x 10 <sup>9</sup>	<b>Exchange Rate</b> 59.13 USD
<b>Chainwork</b> 376.73 x 10 <sup>24</sup> hashes	<b>Circulating Supply</b> 19,056,325 BSV		

More details on [WhatsOnChain](#)

Powered By [TAAI](#)

Based on the selected options, a URL and the HTML code are generated for the integration into any website or application.

## Url

```
https://whatsonchain.com/?theme=light&hideHeader=true&hideFooter=true&hideSearch=true&hideLatestBlocks=true&hideBlockList=true&hideStatsSummary=true&hideNavTools=true&hideTxSummary=true
```

## HTML

```
<iframe src="https://whatsonchain.com/?theme=light&hideHeader=true&hideFooter=true&hideSearch=true&hideLatestBlocks=true&hideBlockList=true&hideStatsSummary=true&hideNavTools=true&hideTxSummary=true" width="100%" height="100%"/>
```

# WoC Plugins

It is our goal to make it easier for anyone to provide metadata or showcase the on-chain data. We believe that by empowering the community, we will continue to unlock another aspect of the boundless potential of the Bitcoin blockchain.

To facilitate this, we have provided the ability to create and publish plugins. The plugins feature allows anyone to add more context or provide a different view/UI for transactions, blocks, addresses, OP\_RETURN data, or user's searched terms.

To create a new plugin or see the existing plugins, navigate to the [WhatsOnChain.com](#) "Tools" Menu and click on "Manage Plugins"

alt text

On the Manage Plugins page, click on the "Lab" tab and fill in the required plugin details.

The screenshot shows the 'Add New Plugin' interface. At the top, there are three tabs: 'Plugins' (selected), 'Lab' (highlighted in blue), and 'Store'. Below the tabs, a message says 'Plugin in my local storage' with a '+' icon. The main area has a header 'Add New Plugin' with a progress bar showing steps 1 (Plugin Details), 2 (Monetize - Coming soon), and 3 (Publish). The 'Plugin Details' step contains fields for 'IconUrl (size 128x128px)' (with a placeholder box), 'Webhook URL - Available Placeholders: {network}, {tx} and {voutindex}' (with an example box), 'Transaction hash for preview' (empty box), 'Vout index for preview' (empty box), and a 'Save' button. To the left, under 'Plugin Type', a dropdown menu is set to 'Data'. Under 'Network Support', checkboxes are checked for 'Mainnet' and 'Testnet', while 'STN' is unchecked. There are also fields for 'Name' and 'Website'.

Below is the list and purpose of different plugin types. And here you can find a [GitHub repo of example plugins](#).

**Data:** You can create this Plugin type to decode and display the OP\_RETURN output. The response provided by the plugin's webhook will be rendered when a user tries to decode the OP\_RETURN data on the transactions page.

**Block:** This type of plugin can be used to share any metadata/stats for a given block. Appears as a new tab on the block page with the data provided by the plugin's webhook.

**Address:** This type of plugin can be used to share any metadata/stats for a given address. Appears as a new tab on the address page with the data provided by the plugin's webhook.

**Transaction:** This type of plugin can be used to share any metadata/stats for a given transaction. Appears as a new tab on the transaction page with the data provided by the plugin's webhook.

**Script:** This type of plugin can be used to share any metadata/stats for a given scripthash. Appears as a new tab on the scripthash page with the data provided by the plugin's webhook.

**Search engine:** If you are interested in sharing data from your specialized indexer, use this plugin type to help users search your indexed data when they search anything on whatsonchain.com via the search bar.

Don't forget to press "Save" after providing the required plugin details.

To Edit, Delete or Publish a plugin, click on the menu option next to the plugin name.

#### Publish Plugin

When a new plugin is created or is in the development phase, it is listed in the "Lab" tab. Plugins listed in the "Lab" are not available publicly and the plugin data is saved locally in the browser's local storage.

All the published plugins are listed under the "Plugins" section.

If you have any questions, please free free to ask via the [WoC devs telegram channel](#).

# On-Chain Data

## Get OP\_RETURN Data by Tx Hash

This endpoint returns OP\_RETURN data as hex for each output in the transaction.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tx/<hash>/opreturn
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

### Response JSON structure

```
[  
  {  
    "n": 2,  
    "hex": "006a075354554b2e434f"  
  }  
]
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
hash	The desired TX hash.

# Output Tags

WhatsOnChain tags recognized transaction outputs in every block. We also provide basic block stats around those tags, both in the [API](#) and on the [Explorer website](#).

If you want your transaction outputs to be tagged, please reach out to us over [email](#) or in our [Telegram channel](#) with these details:

- IDs of a few example transactions
- Any pattern that could be used to detect your transactions with enough precision
- Name of the protocol used for the data/script (if any)
- Brand colors, for tag background and tag text
- The logo file

Optional:

- A link to the webpage for your app, protocol, or token.
- A short description of your app, protocol, or token.

After we've received this information, we can go and add the new tag to our system, which should be available from then on.

# Stats

## Get Block Stats by Height

This endpoint retrieves the block stats for a given height.

- Exchange rate information is not available for blocks processed before 2018/11/19.
- Unidentified block miners are tagged as an empty string.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/main/block/height/<height>/stats
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/b
```

Response JSON structure

## URL Parameters

Parameter	Description
network	The selected network: main.
height	The height of the block to retrieve.

## Get Block Stats by Hash

This endpoint retrieves the block stats for a given hash.

- Exchange rate information is not available for blocks processed before 2018/11/19.
- Unidentified block miners are tagged as an empty string.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/main/block/hash/<hash>/stats
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/b
```

Response JSON structure

## URL Parameters

Parameter	Description
network	The selected network: main.
hash	The hash of the block to retrieve.

## Get Miner Block Stats

This endpoint retrieves the miner block stats for a specified number of days.

Unidentified block miners are tagged as an empty string.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/main/miner/blocks/stats?days={day}
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/m
```

Response JSON structure

```
[  
  {  
    "header": {  
      "size": 297234612,  
      "height": 757673,  
      "time": 1663491738  
    },  
    "stats": {  
      "total_size": 297234612,  
      "total_fee": 23803452  
    },  
    "details": {  
      "miner_tag": "taal.com"  
    }  
  },  
  ...  
]
```

### URL Parameters

Parameter	Description
network	The selected network: main.
days	The number of days to retrieve the data for. Only 1 or 30 days can be selected.

## Get Miner Minimum Fee Rate Stats

This endpoint retrieves the miner minimum fee rate stats for a specified time period.

Minimum fee rate is in sat/KB.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/main/miner/fees?from={from}&to={t
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/m
```

### Response JSON structure

```
[  
  {  
    "miner": "taal.com",  
    "min_fee_rate": 1  
  },  
  {  
    "miner": "CUVVE",  
    "min_fee_rate": 1.0003060717198577  
  },  
  {  
    "miner": "GorillaPool.com 🖱",  
    "min_fee_rate": 1.001001001001001  
  },  
  {  
    "miner": "qdlnk",  
    "min_fee_rate": 1.0001427259380782  
  }]
```

## URL Parameters

Parameter	Description
network	The selected network: main.
from	Starting from as Unix timestamp.
to	Until to as Unix timestamp.

## Get Miner Summary Stats

This endpoint retrieves the miner summary stats for specified days over a 24 hour period.

- Unidentified block miners are tagged as an empty string.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/main/miner/summary/stats?days={da
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/m
```

### Response JSON structure

```
[  
 {  
   "period": 1658275200,  
   "stats": {  
     "": {  
       "count": 3,  
       "header": {  
         "size": 520  
       },  
       "stats": {  
         "total_size": 520,  
         "total_fee": 63057  
       }  
     },  
     "GorillaPool.com 🐒": {  
       "count": 1,  
       "header": {  
         "size": 148698760  
       },  
       "stats": {  
         "total_size": 148698760,  
         "total_fee": 6068680  
       }  
     },  
     "Mining-Dutch": {  
       "count": 2,  
       "header": {  
         "size": 1072828  
       },  
       "stats": {  
         "total_size": 1072828,  
         "total_fee": 789414  
       }  
     },  
     "SBICrypto.com": {  
       "count": 30,  
       "header": {  
         "size": 28091831  
       },  
       "stats": {  
         "total_size": 28091831,  
         "total_fee": 21344586  
       }  
     },  
     "SVPool": {  
       "count": 9,  
       "header": {  
         "size": 480750592  
       }  
     },  
   }  
 ]
```

```
        },
        "stats": {
            "total_size": 480750592,
            "total_fee": 44484376
        }
    },
    "qdlnk": {
        "count": 21,
        "header": {
            "size": 1409145367
        },
        "stats": {
            "total_size": 1409145367,
            "total_fee": 74006627
        }
    },
    "taal.com": {
        "count": 17,
        "header": {
            "size": 978720432
        },
        "stats": {
            "total_size": 978720432,
            "total_fee": 51728556
        }
    }
},
...
]
```

## URL Parameters

Parameter	Description
network	The selected network: main.
days	The number of days to retrieve the data for. Only 90 or 365 days can be selected.

## Get Tag Count by Height

This endpoint retrieves stats on tag count for a specific block at a given height.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/main/block/tagcount/height/{height}
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/b...
```

## Response JSON structure

```
{  
  "count": 9128,  
  "results": [  
    {  
      "name": "certihash",  
      "count": 5588  
    },  
    {  
      "name": "my2cents",  
      "count": 2320  
    },  
    {  
      "name": "run#cryptofights",  
      "count": 709  
    },  
    {  
      "name": "OP_RETURN",  
      "count": 152  
    },  
    {  
      "name": "bitcom#B",  
      "count": 124  
    },  
    {  
      "name": "bitcom#D",  
      "count": 123  
    },  
    {  
      "name": "peergame.com",  
      "count": 29  
    },  
    {  
      "name": "Badge",  
      "count": 17  
    },  
    {  
      "name": "nonstandard",  
      "count": 15  
    },  
    {  
      "name": "run",  
      "count": 12  
    },  
    {  
      "name": "STAS",  
      "count": 11  
    },  
    {  
    }
```

```
        "name": "run#relayx.io",
        "count": 6
    },
    {
        "name": "tdxp.app",
        "count": 5
    },
    {
        "name": "bitcom",
        "count": 4
    },
    {
        "name": "bitcom#twetch",
        "count": 4
    },
    {
        "name": "haste arcade",
        "count": 4
    },
    {
        "name": "metanet#metaid",
        "count": 3
    },
    {
        "name": "DotWallet",
        "count": 2
    }
]
```

## URL Parameters

Parameter	Description
network	The selected network: main.
height	The height of the block.

# WebSockets (Beta)

We recommend the centrifuge websocket client side libraries below:

- [centrifuge-js](#) – for a browser, NodeJS and React Native
- [centrifuge-go](#) - for Go language
- [centrifuge-mobile](#) - for iOS/Android with `centrifuge-go` as basis and [gomobile](#)
- [centrifuge-dart](#) - for Dart and Flutter
- [centrifuge-swift](#) – for native iOS development
- [centrifuge-java](#) – for native Android development and general Java

Non-centrifuge clients connect to separate endpoints specified in each section.

The supported networks are:

- Mainnet
- Testnet

Simple js example using centrifuge client:

```
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/
<script type="text/javascript" src="https://cdn.jsdelivr.net/gh/centrifuge/centrifuge.js@v2.1.0/dist/centrifuge.min.js">
$(function () {
  const centrifuge = new Centrifuge('wss://socket.whatsonchain.com/');

  centrifuge.on('publish', function(message) {
    console.log('Data: ' + JSON.stringify(message.data, null, 2));
  });

  centrifuge.on('disconnect', function(ctx) {
    console.log('Disconnected: ' + ctx.reason + (ctx.reconnect ? ' (reconnecting)' : ''));
  });

  centrifuge.on('connect', function(ctx) {
    console.log('Connected with client ID ' + ctx.client + ' over ' + ctx.transport);
  });

  centrifuge.connect();
});
```

Simple js example using websocket API client

```
<script type="text/javascript">
    window.addEventListener('load', function() {
        let clientID;
        let reconnect = true;
        let numFailures = 0;

        function connect() {
            // Subscribe to multiple channels
            const ws = new WebSocket('wss://socket.whatsonchain.com/w
            ws.onopen = function(e) {
                numFailures = 0;
                console.log("websocket: connection open", e);
                // This is required to trigger connect on server side
                ws.send(JSON.stringify({}));
            };

            ws.onerror = function(e) {
                numFailures++;
                console.log("websocket: connection error: ", e);
            };

            ws.onclose = function (e) {
                if (!reconnect) {
                    return;
                }
                setTimeout(function () {
                    console.log("websocket: connection reconnecting")
                    connect();
                }, Math.min(numFailures * 1000, 20000));
            }

            ws.onmessage = function(e) {
                console.log("websocket: new message");
                processData(e.data);
            };
        }

        connect();

        function processData(rawData) {
            if (rawData.length === 0) {
                console.log("--> ping");
                return;
            }
            console.log("--> " + rawData);
            const data = JSON.parse(rawData);
            const pushType = data.type || 0;
            switch (pushType) {
```

```
----->---->
    case 0:
        console.log("new data from a channel " + data.channel);
        break;
    case 6:
        clientID = data.data.client;
        let subscriptions = [];
        const subs = data.data.subs;
        if (subs) {
            for (const m in subs) {
                if (subs.hasOwnProperty(m)) {
                    subscriptions.push(m);
                }
            }
        }
        console.log("connected with client ID " + clientID);
        break;
    case 7:
        clientID = null;
        if (!data.data.reconnect) {
            reconnect = false;
            ws.close();
            console.log("disconnected from a server, won't reconnect");
        } else {
            console.log("disconnected from a server, will reconnect");
        }
        break;
    case 3:
        console.log("unsubscribed from a channel " + data.channel);
        break;
    case 5:
        console.log("subscribed to a channel " + data.channel);
        break;
    default:
        console.log("unsupported push type " + data.type);
    }
}

});
</script>
```

## New Block Header Event

Receive new block header events in real time.

### URL

- Mainnet:

```
wss://socket.whatsonchain.com/websocket?channels=woc:blockHeader
```

- Testnet:

```
wss://socket-testnet.whatsonchain.com/websocket?  
channels=woc:blockHeader
```

## URL - for centrifuge clients

- Mainnet: `wss://socket.whatsonchain.com/blockheaders`

- Testnet: `wss://socket-testnet.whatsonchain.com/blockheaders`

Data JSON structure



```
        "time": 1614534798,  
        "blocktime": 1614534798  
    }  
}
```

# Block Headers History

Stream block headers starting at a pre-specified block height.

## URL

- Mainnet:

```
wss://socket.whatsonchain.com/websocket/history?from=<blockHeight>
```

- Testnet:

```
wss://socket-testnet.whatsonchain.com/websocket/history?  
from=<blockHeight>
```

## URL - for centrifuge clients

- Mainnet:

```
wss://socket.whatsonchain.com/blockheaders/history?from=<blockHeight>
```

- Testnet:

```
wss://socket-testnet.whatsonchain.com/blockheaders/history?  
from=<blockHeight>
```

Data JSON structure

# Block Transactions

Stream transactions from a block height and transaction index.

- *hex*, *vin* and *vout* values are not published for message sizes greater than 10MB. Recommended to fetch details for such transactions using these [REST](#) endpoints using the published txid value.

## URL

- Mainnet:

```
wss://socket.whatsonchain.com/websocket/transactions?  
from=<blockHeight>&txIndex={txIndex}&to=<blockHeight>&format={hex |  
json}
```

- Testnet:

```
wss://socket-testnet.whatsonchain.com/websocket/transactions?  
from=<blockHeight>&txIndex={txIndex}&to=<blockHeight>&format={hex |  
json}
```

## URL - for centrifuge clients

- **Mainnet:**

```
wss://socket.whatsonchain.com/block/transactions?  
from=<blockHeight>&txIndex={txIndex}&to=<blockHeight>&format={hex |  
json}
```

- **Testnet:**

```
wss://socket-testnet.whatsonchain.com/block/transactions?  
from=<blockHeight>&txIndex={txIndex}&to=<blockHeight>&format={hex |  
json}
```

Data JSON structure

```
{  
  "txid": "20ac30f5aa6aebc0f9e532f32a9691bf7faa332c755d6aea1fd3f6ba36d195  
  "hash": "20ac30f5aa6aebc0f9e532f32a9691bf7faa332c755d6aea1fd3f6ba36d195  
  "size": 219,  
  "version": 2,  
  "locktime": 0,  
  "vin": [  
    {  
      "txid": "15b416adf871f9285788ebf36b67714d9e49f413f3a3f083d80e01535b  
      "vout": 1,  
      "scriptSig": {  
        "asm": "304402204ac1d9f22f09044f66aa6055835b19d5617c19ef180422683  
        "hex": "47304402204ac1d9f22f09044f66aa6055835b19d5617c19ef1804226  
      },  
      "sequence": 4294967295  
    }  
  ],  
  "vout": [  
    {  
      "value": 0,  
      "n": 0,  
      "scriptPubKey": {  
        "asm": "OP_FALSE OP_RETURN 39366665313633626131303138386636",  
        "hex": "006a1039366665313633626131303138386636",  
        "type": "nulldata",  
        "isTruncated": false  
      }  
    },  
    {  
      "value": 0.00007511,  
      "n": 1,  
      "scriptPubKey": {  
        "asm": "OP_DUP OP_HASH160 966a0c69734d51307ba0bcfc9092b51addfaf7  
        "hex": "76a914966a0c69734d51307ba0bcfc9092b51addfaf7888ac",  
        "reqSigs": 1,  
        "type": "pubkeyhash",  
        "addresses": [  
          "1EiKQ9qiAkjDn86JvUt6aJpZvETq8uim2D"  
        ],  
        "isTruncated": false  
      }  
    }  
  ]  
}
```

## URL Parameters

Parameter	Description
blockHeight	The height of the block.
txIndex	Index of a transaction as a starting point in the first block to deliver. Default value 0.
format	<i>Hex or json.</i> If <i>hex</i> , the response will include transaction hex and metadata. If <i>json</i> , <i>hex</i> is excluded. Default value is <i>json</i> .

## Mempool Transactions

Receive mempool transactions in real time.

- Mempool transactions are collected from multiple nodes.
- Client side should be idempotent because duplicate transaction events are not guaranteed to have been removed.
- *hex*, *vin* and *vout* values are not published for message sizes greater than 10MB. Recommended to fetch details for such transactions using these [REST endpoints](#) using the published txid value.

### URL

- Mainnet:

```
wss://socket.whatsonchain.com/websocket?channels=woc:mempoolTx?
filter=<filter1,filter2,...>
```

- Testnet:

```
wss://socket-testnet.whatsonchain.com/websocket?
channels=woc:mempoolTx?filter=<filter1,filter2,...>
```

### URL - for centrifuge clients

- Mainnet:

```
wss://socket.whatsonchain.com/mempool?filter=<filter1,filter2,...>
```

- Testnet:

```
wss://socket-testnet.whatsonchain.com/mempool?
filter=<filter1,filter2,...>
```

## Data JSON structure

```
{  
    "hex": "02 ...",  
    "txid": "20ac30f5aa6aebc0f9e532f32a9691bf7faa332c755d6aea1fd3f6ba36d195  
    "hash": "20ac30f5aa6aebc0f9e532f32a9691bf7faa332c755d6aea1fd3f6ba36d195  
    "size": 219,  
    "version": 2,  
    "locktime": 0,  
    "vin": [  
        {  
            "txid": "15b416adf871f9285788ebf36b67714d9e49f413f3a3f083d80e01535b  
            "vout": 1,  
            "scriptSig": {  
                "asm": "304402204ac1d9f22f09044f66aa6055835b19d5617c19ef180422683  
                "hex": "47304402204ac1d9f22f09044f66aa6055835b19d5617c19ef1804226  
            },  
            "sequence": 4294967295  
        }  
    ],  
    "vout": [  
        {  
            "value": 0,  
            "n": 0,  
            "scriptPubKey": {  
                "asm": "OP_FALSE OP_RETURN 39366665313633626131303138386636",  
                "hex": "006a1039366665313633626131303138386636",  
                "type": "nulldata",  
                "isTruncated": false  
            }  
        },  
        {  
            "value": 0.00007511,  
            "n": 1,  
            "scriptPubKey": {  
                "asm": "OP_DUP OP_HASH160 966a0c69734d51307ba0bcfc9092b51addfaf7  
                "hex": "76a914966a0c69734d51307ba0bcfc9092b51addfaf7888ac",  
                "reqSigs": 1,  
                "type": "pubkeyhash",  
                "addresses": [  
                    "1EiKQ9qiAkjDn86JvUt6aJpZvETq8uim2D"  
                ],  
                "isTruncated": false  
            }  
        }  
    ]  
}
```

## URL Parameters

Parameter	Description
filter	The filter transactions based on the output ( <i>vout</i> ) types. The available filters are: <i>nulldata</i> , <i>multisig</i> , <i>pubkey</i> , <i>pubkeyhash</i> , <i>scripthash</i> , and <i>nonstandard</i> .

## Confirmed Transactions (Unavailable)

This feature is currently unavailable as we are working on a v2. Please let us know in our [Telegram channel](#) if you are interested in streaming and what features you would like to see in the next version.

Receive confirmed transactions in real time.

- Confirmed transactions are collected from multiple nodes.
- Client side should be idempotent because duplicate transaction events are not guaranteed to have been removed.
- hex*, *vin* and *vout* values are not published for transaction sizes greater than 1 MB. Recommended to fetch details for such transactions using these [REST endpoints](#) using the published txid value.

### URL

- Mainnet:

```
wss://socket.whatsonchain.com/websocket?channels=woc:confirmedTx?
filter=<filter1,filter2,...>
```

- Testnet:

```
wss://socket-testnet.whatsonchain.com/websocket?
channels=woc:confirmedTx?filter=<filter1,filter2,...>
```

### URL - for centrifuge clients

- Mainnet:

```
wss://socket.whatsonchain.com/confirmed?filter=<filter1,filter2,...>
```

- **Testnet:**

```
wss://socket-testnet.whatsonchain.com/confirmed?  
filter=<filter1,filter2,...>
```

Data JSON structure

```
{  
    "hex": "01 ...",  
    "txid": "651dd0bc40520dfc5927b8b9a6fbb0d3d7ff7314fae28d6c5ddad66e1f75ff",  
    "hash": "651dd0bc40520dfc5927b8b9a6fbb0d3d7ff7314fae28d6c5ddad66e1f75ff",  
    "size": 442,  
    "version": 1,  
    "locktime": 0,  
    "vin": [  
        {  
            "txid": "719f1f2e79c29f30d60b106518f29abca0e3d17fc79731c7bc3750413",  
            "vout": 2,  
            "scriptSig": {  
                "asm": "30450221008117adcf9c88330b85efa6772ba6b8cecccc473ecea8160",  
                "hex": "4830450221008117adcf9c88330b85efa6772ba6b8cecccc473ecea81"},  
            "sequence": 4294967295  
        },  
        {"vout": [  
            {  
                "value": 0.00003822,  
                "n": 0,  
                "scriptPubKey": {  
                    "asm": "OP_DUP OP_HASH160 09cc4559bdcb84cb35c107743f0dbb10d66679c",  
                    "hex": "76a91409cc4559bdcb84cb35c107743f0dbb10d66679cc88ac",  
                    "reqSigs": 1,  
                    "type": "pubkeyhash",  
                    "addresses": [  
                        "1tonicZQwN2BNKhVwPXqh8ez3q56y1EYw"  
                    ],  
                    "isTruncated": false  
                },  
                {"value": 0.00038224,  
                "n": 1,  
                "scriptPubKey": {  
                    "asm": "OP_DUP OP_HASH160 b8cba27cbf0357e1e74ede6b4ac17f49aa29799",  
                    "hex": "76a914b8cba27cbf0357e1e74ede6b4ac17f49aa29799188ac",  
                    "reqSigs": 1,  
                    "type": "pubkeyhash",  
                    "addresses": [  
                        "1Hr7H5vdaxordBEu9CJxnfD2NNWEnwEfih"  
                    ],  
                    "isTruncated": false  
                }  
            }  
        ]  
    }  
}
```

```
...  
],  
"blockhash": "0000000000000000000045876574d74d4006b8b821525135720a0a12343  
"confirmations": 1,  
"time": 1616419527,  
"blocktime": 1616419527,  
"blockheight": 679542  
}
```

## URL Parameters

Parameter	Description
filter	The filter transactions based on the output ( <i>vout</i> ) types. The available filters are: <i>nulldata</i> , <i>multisig</i> , <i>pubkey</i> , <i>pubkeyhash</i> , <i>scripthash</i> , and <i>nonstandard</i> .

# Chain Stats

Receive common chain stats every 10 seconds.

## URL

- Mainnet:

```
wss://socket.whatsonchain.com/websocket?channels=woc:chainStats
```

- Testnet:

```
wss://socket-testnet.whatsonchain.com/websocket?  
channels=woc:chainStats
```

## URL - for centrifuge clients

- Mainnet: <wss://socket.whatsonchain.com/stats>
- Testnet: <wss://socket-testnet.whatsonchain.com/stats>

Data JSON structure

## Examples

# Customized Events

If your application needs customized/filtered events via WebSockets, please let us know in the [WoC devs telegram channel](#).

# Tokens

# 1Sat Ordinals (Beta)

1SatOrdinals is an implementation of Ordinals running on the BSV blockchain. More information [here](#).



At this moment only NFT tokens are supported. BSV-20/BSV-21 fungible tokens are not supported yet, but will be in the near future.

## Get Token by Origin Outpoint

The token origin represents the moment the NFT was created on the blockchain (more details [here](#)).

This endpoint retrieves the token details by origin outpoint (the outpoint is the transaction ID and the output index).

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/token/1satordinals/<ori
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

Response JSON example

```
{  
  "token": {  
    "outpoint": "b847a5c9aff596fb98435dd179fd7a6d541d54556518c357f53f54  
    "scriptHash": "e260c235d47f4418076ad4de93942508d0d7de6286337d753ee5  
    "ownerAddress": "1siPrNPYBmWEwgsJLu6LS84iqJYBSCAf3",  
    "origin": {  
      "number": "827748:753:0",  
      "outpoint": "b847a5c9aff596fb98435dd179fd7a6d541d54556518c357f53  
      "scriptHash": "e260c235d47f4418076ad4de93942508d0d7de6286337d753  
      "ownerAddress": "1siPrNPYBmWEwgsJLu6LS84iqJYBSCAf3",  
      "data": {  
        "insc": {  
          "file": {  
            "hash": "Yb51ss8Unqt43ziwJUm8LNCKSooBWrJLe5x2A/qldEw=",  
            "size": 13,  
            "type": "text/plain"  
          },  
          "text": "464400.bsvmap",  
          "words": [  
            "bsvmap",  
            "464400"  
          ]  
        },  
        "types": [  
          "text"  
        ]  
      }  
    }  
  },  
  "data": {  
    "insc": {  
      "file": {  
        "hash": "Yb51ss8Unqt43ziwJUm8LNCKSooBWrJLe5x2A/qldEw=",  
        "size": 13,  
        "type": "text/plain"  
      },  
      "text": "464400.bsvmap",  
      "words": [  
        "bsvmap",  
        "464400"  
      ]  
    },  
    "types": [  
      "text"  
    ]  
  },  
  "current": {  
    "txid": "b847a5c9aff596fb98435dd179fd7a6d541d54556518c357f53f548  
  }
```

{  
}

## URL Parameters

Parameter	Description
network	The selected network: main or test.
origin outpoint	Inscription ID. More info <a href="#">here</a> .

## Get Token by Origin Number

The token origin represents the moment the NFT was created on the blockchain (more details [here](#)).

This endpoint retrieves the token details by Origin number (the number is the block height, transaction index, and output index).

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/token/1satordinals/<ord
```

### Response JSON structure

```
{  
  "token": {  
    "outpoint": string,  
    "scriptHash": string,  
    "ownerAddress": string,  
    "origin": {  
      "number": string,  
      "outpoint": string,  
      "scriptHash": string,  
      "ownerAddress": string,  
      "data": {  
        "insc": {  
          "file": {  
            "hash": string,  
            "size": integer,  
            "type": string  
          },  
          "text": string,  
          "words": [  
            string,  
            string  
          ]  
        },  
        "types": [  
          string  
        ]  
      }  
    }  
  },  
  "data": {  
    "insc": {  
      "file": {  
        "hash": string,  
        "size": integer,  
        "type": string  
      },  
      "text": string,  
      "words": [  
        string  
      ]  
    },  
    "types": [  
      string  
    ]  
  },  
  "current": {  
    "txid": string  
  }  
}
```

{}

**Example:**

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

Response JSON example

```
{  
  "token": {  
    "outpoint": "b847a5c9aff596fb98435dd179fd7a6d541d54556518c357f53f54  
    "scriptHash": "e260c235d47f4418076ad4de93942508d0d7de6286337d753ee5  
    "ownerAddress": "1siPrNPYBmWEwgsJLu6LS84iqJYBSCAf3",  
    "origin": {  
      "number": "827748:753:0",  
      "outpoint": "b847a5c9aff596fb98435dd179fd7a6d541d54556518c357f53  
      "scriptHash": "e260c235d47f4418076ad4de93942508d0d7de6286337d753  
      "ownerAddress": "1siPrNPYBmWEwgsJLu6LS84iqJYBSCAf3",  
      "data": {  
        "insc": {  
          "file": {  
            "hash": "Yb51ss8Unqt43ziwJUm8LNCKSooBWrJLe5x2A/qldEw=",  
            "size": 13,  
            "type": "text/plain"  
          },  
          "text": "464400.bsvmap",  
          "words": [  
            "bsvmap",  
            "464400"  
          ]  
        },  
        "types": [  
          "text"  
        ]  
      }  
    }  
  },  
  "data": {  
    "insc": {  
      "file": {  
        "hash": "Yb51ss8Unqt43ziwJUm8LNCKSooBWrJLe5x2A/qldEw=",  
        "size": 13,  
        "type": "text/plain"  
      },  
      "text": "464400.bsvmap",  
      "words": [  
        "bsvmap",  
        "464400"  
      ]  
    },  
    "types": [  
      "text"  
    ]  
  },  
  "current": {  
    "txid": "b847a5c9aff596fb98435dd179fd7a6d541d54556518c357f53f548  
  },  
}
```

{  
}  
}

## URL Parameters

Parameter	Description
network	The selected network: main or test.
origin number	Origin number. More info <a href="#">here</a> .

## Get Token's Data Content

This endpoint retrieves the content of the token's file, in case it has one.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/token/1satordinals/<out
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

### Response Plain text

```
464400.bsvmap
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
outpoint	Inscription ID. More info <a href="#">here</a> .

## Get the Latest Token's Transfer

This endpoint retrieves the token's latest transfer with the origin token.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/token/1satordinals/<out
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

Response JSON example

```
{  
  "token": {  
    "outpoint": "7e796f68d56b7b530b91ff551f402d8972aa3ea6ecf9eb7187616c  
    "scriptHash": "c34e301999f898cb3e8f50bc85b5f1c3a45ab3aeee33f125c5978  
    "scriptLockType": "p2pkh",  
    "ownerAddress": "1N1QG9hdaMhaTwBDweVrrkbyrXxNeCBHst",  
    "origin": {  
      "number": "783989:21395:156",  
      "outpoint": "6fec063b0ee00489f79a0e9c0793e0bcb265f81faa43e36193  
      "vout": 156,  
      "scriptHash": "28110cffb0f31f2ea4d13cdb9a848ffd0e3cd912ebf3ab523  
      "scriptLockType": "nonstandard",  
      "ownerAddress": "1J3CFrgrmqmHrEtqCiP6ybqedKBwA3tELN",  
      "spentTxid": "fda8480b34526ed3b5e780288b6b3851de94ef81018e3e752b  
      "data": {  
        "inSc": {  
          "file": {  
            "hash": "Te0qczWHftnhDJjMza6Z/siyzLOTN9QkZ71hYVPW/mQ=",  
            "size": 1841,  
            "type": "image/png"  
          }  
        },  
        "map": {  
          "app": "BigBlockPunkz",  
          "collection": "BigBlockPunkz",  
          "type": "ord"  
        },  
        "types": [  
          "file"  
        ]  
      }  
    },  
    "current": {  
      "txid": "7e796f68d56b7b530b91ff551f402d8972aa3ea6ecf9eb7187616cb",  
      "txIndex": 2553,  
      "blockHeight": 832795,  
      "blockTime": 1708693512  
    }  
  }  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
outpoint	Inscription ID. More info <a href="#">here</a> .

## Get the Token's Transfer History

This endpoint retrieves the token's history of transfers.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/token/1satordinals/<out
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

Response JSON example

```
{  
  "tokens": [  
    {  
      "outpoint": "6fecd063b0ee00489f79a0e9c0793e0bcb265f81faa43e36193  
      "vout": 156,  
      "scriptHash": "28110cffb0f31f2ea4d13cdb9a848ffd0e3cd912ebf3ab523  
      "scriptLockType": "nonstandard",  
      "ownerAddress": "1J3CFrgrmqmHrEtqCiP6ybqedKBwA3tELN",  
      "spentTxid": "fda8480b34526ed3b5e780288b6b3851de94ef81018e3e752b  
      "origin": {  
        "number": "783989:21395:156",  
        "outpoint": "6fecd063b0ee00489f79a0e9c0793e0bcb265f81faa43e36  
        "vout": 156,  
        "scriptHash": "28110cffb0f31f2ea4d13cdb9a848ffd0e3cd912ebf3ab  
        "scriptLockType": "nonstandard",  
        "ownerAddress": "1J3CFrgrmqmHrEtqCiP6ybqedKBwA3tELN",  
        "spentTxid": "fda8480b34526ed3b5e780288b6b3851de94ef81018e3e7  
        "data": {  
          "insc": {  
            "file": {  
              "hash": "Te0qczWHftnhDJjMza6Z/siyzLOTN9QkZ71hYVPW/mQ",  
              "size": 1841,  
              "type": "image/png"  
            }  
          },  
          "map": {  
            "app": "BigBlockPunkz",  
            "collection": "BigBlockPunkz",  
            "type": "ord"  
          },  
          "types": [  
            "file"  
          ]  
        }  
      },  
      "data": {  
        "insc": {  
          "file": {  
            "hash": "Te0qczWHftnhDJjMza6Z/siyzLOTN9QkZ71hYVPW/mQ=",  
            "size": 1841,  
            "type": "image/png"  
          }  
        },  
        "map": {  
          "app": "BigBlockPunkz",  
          "collection": "BigBlockPunkz",  
          "type": "ord"  
        }  
      }  
    }  
  ]  
}
```

```
        ],
        "types": [
            "file"
        ]
    },
    "current": {
        "txid": "6fecd063b0ee00489f79a0e9c0793e0bcb265f81faa43e361938",
        "txIndex": 21395,
        "blockHeight": 783989,
        "blockTime": 1679320103
    }
},
//...
{
    "outpoint": "7e796f68d56b7b530b91ff551f402d8972aa3ea6ecf9eb71876",
    "scriptHash": "c34e301999f898cb3e8f50bc85b5f1c3a45ab3aee33f125c5",
    "scriptLockType": "p2pkh",
    "ownerAddress": "1N1QG9hdaMhaTwBDweVrrkbyrXxNeCBHst",
    "origin": {
        "number": "783989:21395:156",
        "outpoint": "6fecd063b0ee00489f79a0e9c0793e0bcb265f81faa43e36",
        "vout": 156,
        "scriptHash": "28110cffb0f31f2ea4d13cdb9a848ffd0e3cd912ebf3ab",
        "scriptLockType": "nonstandard",
        "ownerAddress": "1J3CFrgrmqmHrEtqCiP6ybqedKBwA3tELN",
        "spentTxid": "fda8480b34526ed3b5e780288b6b3851de94ef81018e3e7",
        "data": {
            "inSc": {
                "file": {
                    "hash": "Te0qcZWfHftnhDJjMza6Z/siyzLOTN9QkZ71hYVPW/mQ",
                    "size": 1841,
                    "type": "image/png"
                }
            },
            "map": {
                "app": "BigBlockPunkz",
                "collection": "BigBlockPunkz",
                "type": "ord"
            }
        },
        "types": [
            "file"
        ]
    }
},
"current": {
    "txid": "7e796f68d56b7b530b91ff551f402d8972aa3ea6ecf9eb718761",
    "txIndex": 2553,
    "blockHeight": 832795,
    "blockTime": 1708693512
}
```

```
        }
    }
],
"total_count": 9
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
outpoint	Inscription ID. More info <a href="#">here</a> .

## Query Parameters

Parameter	Description
skip	Skip items. For pagination.
limit	Limit number of items. For pagination.

# Get Transaction's Tokens

This endpoint retrieves tokens that a transaction contains.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/token/1satordinals/tx/<
```

### Response JSON structure

```
{  
  "tokens": [  
    {  
      "outpoint": string,  
      "scriptHash": string,  
      "scriptLockType": string,  
      "ownerAddress": string,  
      "spentTxid": string,  
      "origin": {  
        "number": string,  
        "outpoint": string,  
        "vout": integer,  
        "scriptHash": string,  
        "scriptLockType": string,  
        "ownerAddress": string,  
        "spentTxid": string,  
        "data": {  
          "inSc": {  
            "file": {  
              "hash": string,  
              "size": integer,  
              "type": string  
            }  
          },  
          "map": {  
            string: string  
          },  
          "types": [  
            string  
          ]  
        }  
      },  
      "data": {  
        "inSc": {  
          "file": {  
            "hash": string,  
            "size": integer,  
            "type": string  
          }  
        },  
        "map": {  
          string: string  
        },  
        "type": string  
      },  
      "sigma": [  
        {  
          "address": string.  
        }  
      ]  
    }  
  ]  
}
```

```
        "----- : -----",
        "algorithm": string,
        "signature": string,
        "valid": boolean,
        "vin": integer
    }
],
"types": [
    string
]
},
"current":{
    "txid": string,
    "txIndex": integer,
    "blockHeight": integer,
    "blockTime": integer
}
}
],
"total_count": integer
}
```

Example:

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

Response JSON structure

```
{  
  "tokens": [  
    {  
      "outpoint": "9c59baa1944e93771379f110526f7c80a9c9627af583df3f760  
      "scriptHash": "435ef0918227553dec7539da2fd0faeb4dfda2405f46b1219  
      "scriptLockType": "nonstandard",  
      "ownerAddress": "1PjgYHLBqkKhzddfhNN6A428irtAhDWZTN",  
      "spentTxid": "99995013a737ed4437d87a2fbbaafeae89630cd8ea228afc71f  
      "origin": {  
        "number": "783989:21395:156",  
        "outpoint": "6fecd063b0ee00489f79a0e9c0793e0bcb265f81faa43e36  
        "vout": 156,  
        "scriptHash": "28110cffb0f31f2ea4d13cdb9a848ffd0e3cd912ebf3ab  
        "scriptLockType": "nonstandard",  
        "ownerAddress": "1J3CFrgrmqmHrEtqCiP6ybqedKBwA3tELN",  
        "spentTxid": "fda8480b34526ed3b5e780288b6b3851de94ef81018e3e7  
        "data": {  
          "inSc": {  
            "file": {  
              "hash": "Te0qcZWftnhDJjMza6Z/siyzL0TN9QkZ71hYVPW/mQ  
              "size": 1841,  
              "type": "image/png"  
            }  
          }  
        },  
        "map": {  
          "app": "BigBlockPunkz",  
          "collection": "BigBlockPunkz",  
          "type": "ord"  
        },  
        "types": [  
          "file"  
        ]  
      }  
    },  
    {  
      "data": {  
        "inSc": {  
          "file": {  
            "hash": "hTBTPtJh/pEWUQnCmVFdu296UhaslM9q2i1DJtB741M=",  
            "size": 785409,  
            "type": "image/png"  
          }  
        }  
      },  
      "map": {  
        "app": "ZoideNFT",  
        "subType": "collectionItem",  
        "subTypeData": {  
          "collectionId": "2a48f67ae1da8e679210b073d2e2cb5f0eb7bf",  
          "mintNumber": "6".  
        }  
      }  
    }  
  ]  
}
```

```
        "name": "Reverb 6",
        "rarityLabel": "COMMON"
    },
    "type": "ord"
},
"sigma": [
{
    "address": "12KP5KzkBwtsc1UrTrsBCJzgqKn8UqaYQq",
    "algorithm": "BSM",
    "signature": "H6n1PVuBqlTsK2B/Sh+oevNRcZDmTtmkRVTZeRGjn",
    "valid":true,
    "vin":0
},
],
"types": [
    "file"
]
},
"current": {
    "txid": "9c59baa1944e93771379f110526f7c80a9c9627af583df3f760f",
    "txIndex": 829,
    "blockHeight": 832677,
    "blockTime": 1708617885
}
}
],
"total_count": 1
}
```

## URL Parameters

Parameter	Description
network	The selected network: main or test.
txid	Transaction ID.

## Query Parameters

Parameter	Description
skip	Skip items. For pagination.
limit	Limit number of items. For pagination.

# STAS Tokens (Beta)

STAS is a token protocol that consists of a bitcoin script for locking/unlocking bitcoin transaction outputs.

## Get all Tokens

This endpoint retrieves a list of supported tokens along with the relevant metadata.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/tokens
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

Response JSON example

```
{  
  "tokens": [  
    {  
      "description": "-",  
      "image": "https://firebasestorage.googleapis.com/v0/b/musicar  
      "name": "fdsfs",  
      "properties": {  
        "issuer": {  
          "email": "info@musicart.io",  
          "governing_law": "England and Wales",  
          "issuer_country": "UK",  
          "jurisdiction": "UK",  
          "legal_form": "Limited",  
          "organisation": "Calekta Limited"  
        },  
        "legal": {  
          "terms": "\u00a9 2020 TAAL TECHNOLOGIES SEZC\nALL RIG  
        }  
      },  
      "protocol": "STAS",  
      "schema_id": "NFT1.0/MA",  
      "symbol": "ujtVA5",  
      "token_id": "7bfb2ae7e878d107191259c1be72cf116da3f59d"  
    },  
    {  
      "token_id": "2",  
      "protocol": "STAS",  
      "ticker": "CIPS",  
      ....  
    }  
  ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main.

## Get Token by Id

This endpoint retrieves the token metadata including contract and issuance details.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/token/<tokenId>/<symbol>
```

## cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

## Response JSON example

```
{  
    "token": {  
        "contract_txs": [  
            "72ae93f5fdc5a4fa5abfab780d7d1c8ad988d25ae34578b80dce5ff9e5c3  
        ],  
        "description": "generating stas token",  
        "image": "https://firebasestorage.googleapis.com/v0/b/nftdev/o/nf  
        "issuance_txs": [  
            "b00bcc9dfc7dd483ff989f3d1d3f091b23e88ca2207c9e15788e3e84a5b1  
        ],  
        "name": "test",  
        "properties": {  
            "issuer": {  
                "email": "info@vaionex.com",  
                "governing_law": "US",  
                "issuer_country": "US",  
                "jurisdiction": "US",  
                "legal_form": "Limited",  
                "organisation": "vaionex corp."  
            },  
            "legal": {  
                "terms": "test"  
            }  
        },  
        "protocol": "STAS",  
        "schema_id": "NFT1.0/MA",  
        "symbol": "test1124",  
        "token_id": "c402bc341157265f5f8a747c11099847056a229a",  
        "tx_count": 3  
    }  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main.
tokenId	The unique ID of the token.
symbol	The symbol used in the registration of the token.

## Get Token UTXOs for Address

This endpoint retrieves the token UTXOs for a given address.

### HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/token
```

### cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

### Response JSON example

```
{  
  "address": "1JsQieDaYUg5pSRtpeEQhTv3buHvSX7WH4",  
  "utxos": [  
    {  
      "amount": 2000,  
      "index": 1,  
      "redeemAddr": "c402bc341157265f5f8a747c11099847056a229a",  
      "symbol": "test1124",  
      "txid": "cf3968e6950286cf94aa9efb7a1cc1b9c74190d159ff6d2a4dc7  
    }  
  ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main.
address	The address

## Get Address Token Balance

This endpoint retrieves the token balances for a given address.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/address/<address>/token
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/a
```

Response JSON example

```
{  
    "address": "1JsQieDaYUg5pSRtpeEQhTv3buHvSX7WH4",  
    "tokens": [  
        {  
            "balance": 0,  
            "image": "https://firebasestorage.googleapis.com/v0/b/nftdev/  
            "protocol": "STAS",  
            "redeemAddr": "c402bc341157265f5f8a747c11099847056a229a",  
            "symbol": "test2"  
        },  
        {  
            "balance": -2000,  
            "image": "https://firebasestorage.googleapis.com/v0/b/nftdev/  
            "protocol": "STAS",  
            "redeemAddr": "c402bc341157265f5f8a747c11099847056a229a",  
            "symbol": "test1124"  
        }  
    ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main.
address	The address

## Get Token Transactions

This endpoint retrieves the transactions for a given token.

HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/token/<tokenId>/<symbol>
```

cURL

```
curl --location --request GET "https://api.whatsonchain.com/v1/bsv/main/t
```

Response JSON example

```
{  
    "redeem_addr": "c402bc341157265f5f8a747c11099847056a229a",  
    "symbol": "test1124",  
    "txs": [  
        "cf3968e6950286cf94aa9efb7a1cc1b9c74190d159ff6d2a4dc7d9feb6e50c55  
        "b00bcc9dfc7dd483ff989f3d1d3f091b23e88ca2207c9e15788e3e84a5b10c8b  
        "72ae93f5fdc5a4fa5abfab780d7d1c8ad988d25ae34578b80dce5ff9e5c3db03  
    ]  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main.
tokenId	The unique ID of the token.
symbol	The symbol used in the registration of the token.

# Get Token Output Details

This endpoint returns the details of a valid token output or *Not Found (404)* for an invalid token output.

## HTTP Request

```
GET https://api.whatsonchain.com/v1/bsv/<network>/token/tx/<txid>/out/<in
```

## cURL

```
curl --location --request GET "wget -O- https://api.whatsonchain.com/v1/b
```

## Response JSON structure

```
{  
    "has_data": true,  
    "icon_url": "https://firebasestorage.googleapis.com/v0/b/nftdev/o/nft  
    "index": 1,  
    "is_splittable": true,  
    "symbol": "test1124",  
    "token_id": "c402bc341157265f5f8a747c11099847056a229a",  
    "txid": "cf3968e6950286cf94aa9efb7a1cc1b9c74190d159ff6d2a4dc7d9feb6e5  
    "value": 2000  
}
```

## URL Parameters

Parameter	Description
network	The selected network: main.
hash	The hash/txId of the transaction.
index	The Output/Vout index.

# Change Log

These are the most recent changes and additions to the API:

- 12/31/2024
  - Changes
    - Transaction - [Get Merkle Proof](#)
- 12/02/2024
  - New
    - Tokens - [1Sat Ordinals \(Beta\)](#)
- 10/22/2024
  - New
    - Transaction - [Get a Transaction as Binary](#)
- 09/09/2024
  - Changes
    - (Un)Spent Transaction Outputs - [Get Confirmed UTXOs by Address](#)
    - (Un)Spent Transaction Outputs - [Get Confirmed UTXOs by Script](#)
- 10/05/2024
  - Changes
    - Block - [Get Header by Hash or Height](#)
- 19/03/2024
  - New
    - Address - [Get Associated Scripthashes](#)
    - Address - [Bulk History \(Beta\)](#)
- 21/11/2023
  - New
    - (Un)Spent Transaction Outputs - [Bulk Spent Transaction Outputs \(Beta\)](#)
- 15/11/2023
  - New
    - (Un)Spent Transaction Outputs - [Bulk Unconfirmed UTXOs by Address](#)

## (Beta)

- (Un)Spent Transaction Outputs - [Bulk Confirmed UTXOs by Address \(Beta\)](#)
  - (Un)Spent Transaction Outputs - [Bulk Unconfirmed UTXOs by Script \(Beta\)](#)
  - (Un)Spent Transaction Outputs - [Bulk Confirmed UTXOs by Script \(Beta\)](#)
  - Address - [Bulk Unconfirmed Balance \(Beta\)](#)
  - Address - [Bulk Confirmed Balance \(Beta\)](#)
  - Address - [Bulk Unconfirmed History \(Beta\)](#)
  - Address - [Bulk Confirmed History \(Beta\)](#)
  - Script - [Bulk Unconfirmed Script History \(Beta\)](#)
  - Script - [Bulk Confirmed Script History \(Beta\)](#)
- 26/10/2023
    - New
      - (Un)Spent Transaction Outputs - [Get Confirmed UTXOs by Address \(Beta\)](#)
      - (Un)Spent Transaction Outputs - [Get Confirmed UTXOs by Script \(Beta\)](#)
      - (Un)Spent Transaction Outputs - [Get Unconfirmed Spent Tx Output \(Beta\)](#)
      - (Un)Spent Transaction Outputs - [Get Confirmed Spent Tx Output \(Beta\)](#)
      - (Un)Spent Transaction Outputs - [Get Spent Transaction Output \(Beta\)](#)
      - Address - [Get Confirmed Balance \(Beta\)](#)
      - Address - [Get Confirmed History \(Beta\)](#)
      - Script - [Get Confirmed Script History \(Beta\)](#)
  - 28/07/2023
    - New
      - Address - [Get Address Usage Status](#)
      - Script - [Get Script Usage Status](#)
  - 07/07/2023
    - New
      - Address - [Get Mempool Balance](#)

- Address - [Get Mempool History](#)
  - Address - [Get Unspent Mempool Transactions](#)
  - Script - [Get Mempool History](#)
  - Script - [Get Script Mempool Unspent Transactions](#)
- 03/05/2023
    - New
      - On-Chain Data - [Get OP\\_RETURN Data by Tx Hash](#)
  - 20/04/2023
    - New
      - Transaction - [Get Transaction Propagation Status](#)
  - 16/03/2023
    - New
      - [Output Tags](#) (documentation only)
  - 21/02/2023
    - New
      - Block - [Get Header Byte File Links](#)
      - Block - [Get Latest Header Bytes](#)
  - 20/10/2022
    - New
      - Stats - [Get tag count by height](#)
  - 18/10/2022
    - New
      - Stats - [Get miner block stats](#)
      - Stats - [Get miner summary stats](#)
  - 13/10/2022
    - New
      - Stats - [Get block stats by height](#)
      - Stats - [Get block stats by hash](#)
  - 10/10/2022

- New
  - Chain info - [Get peer info](#)
  - Exchange rate - [Get historical exchange rate](#)
- 19/07/2022
  - New
    - [WoC Plugins](#)
- 11/05/2022
  - New
    - [STAS Token endpoints](#)
    - WebSockets - Options to filter [Mempool transactions](#) by the output type.
    - WebSockets - Options to filter [Confirmed transactions](#) by the output type.
    - [WoC Widgets](#)
- 02/03/2022
  - New
    - WebSockets - [Block transactions](#)
- 10/01/2022
  - New
    - [Get chain tips](#)
    - [Get block header by hash with block-headers-client response format](#)
- 03/08/2021
  - New
    - [Bulk transactions status](#)
    - [Bulk transactions raw data](#)
- 17/06/2021
  - New
    - WebSockets - Adding testnet support
- 14/06/2021
  - New
    - WebSockets - [Chain Stats](#)

- WebSockets - Extending support for non-centrifuge [clients](#)
- 22/03/2021
  - New
    - WebSockets - [Mempool transactions](#)
    - WebSockets - [Confirmed transactions](#)
    - WebSockets - [Customized events](#)
- 02/03/2021
  - Changes
    - Deprecating WoC Merchant API. Recommended to switch to [TAAL Merchant API](#).
    - Deprecating WoC Bulk Broadcast. Recommended to switch to [TAAL Merchant API](#).
  - New
    - Adding [TAAL Merchant API details](#)
    - WebSockets - [New block header event](#)
    - WebSockets - [Block headers history](#)
- 26/11/2020
  - New
    - [Bulk script unspent transactions](#)
    - [Bulk address unspent transactions](#)
    - [Bulk address balance](#)
- 13/07/2020
  - New
    - [Get block header by hash](#)
    - [Get last 10 block headers](#)
- 25/05/2020
  - New
    - Merchant API - [Get Fee Quotes](#) from multiple transaction processors
    - Merchant API - [Submit tx](#) to a transaction processor
    - Merchant API - [Request Transaction status](#) from a transaction processor

- 08/03/2020
  - New
    - Enabled [Get raw transaction data \(hex\)](#) for testnet
    - Enabled [Get raw transaction output data by index \(hex\)](#) for testnet
    - [Get script history](#)
    - [Get script unspent transactions](#)
    - [Get exchange rate](#)
- 03/02/2020
  - Changes
    - Removing support of STN network from address balance, history and UTXOs endpoints
    - Removing hex field from response body of [Get transaction by hash](#) request
    - Limiting vout[x].scriptPubKey.asm and vout[x].scriptPubKey.hex size to return max 100KB in response body of [Get transaction by hash](#)
    - Adding vout[x].scriptPubKey.isTruncated flag, if data size is more than 100KB in vout[x].scriptPubKey.hex
    - Removing old merkle proof endpoint \*/tx/{txid}/merkleproof
  - New
    - [Get raw transaction data](#)
    - [Get raw transaction output data by index \(hex\)](#)
    - [Get Merkle Proof for transaction \(Beta\)](#)
- 18/01/2020
  - New
    - [Bulk transaction details](#)
    - [Get circulating supply](#)
    - [Community Libraries](#)
- 18/11/2019
  - New
    - Bulk Broadcast transactions
- 09/11/2019

- New
  - [Get merkle proof](#)
- 21/10/2019
  - New
    - [Decode raw transaction](#)
- 23/06/2019
  - New
    - [Download address statement](#)
    - [Download transaction receipt](#)
  - Fixed
    - [Broadcast transaction](#) endpoint now returns txid on success or error msg from node on failure
- 19/05/2019
  - New
    - [Get address balance](#)
    - [Get address history](#)
    - [Get address UTXOs](#)
    - [Get explorer links](#)
- 22/04/2019
  - New
    - [Get chain info](#)
    - [Get mempool info](#)
    - [Get mempool transactions](#)
    - [Get address info](#)
- 15/04/2019
  - New
    - Get Health
    - [Get block by hash](#)
    - [Get block by height](#)
    - [Get block pages](#)

- [Get transactions by hash](#)
- [Broadcast transactions](#)

# Community Libraries

Awesome community-maintained libraries

```
 .---.      .---.      .---.
 .---|__|      .-.      |~~~|
 .--|==| B |_      |_|      |~~~|--.
 |  |==| I |'\'      .---!~|  .--|  |--| | | | |
 |  |  | T |.'\'      |==|=| --|%%| W |  |
 |  |  | C |\.\'      | S | |__|  | O |py|
 |go|  | O | \  \      |==|=| ==|  | C |  |
 |  |  | I |  \.'\'  | V |_|__|  |~~~|__|
 |  |==| N |  \.'\'|==|=| ~| --|%%|~~~|--|
 ^--^---^---^      `---^---^---^---^---^---'
```

## Golang

Description	Link
Go Wrapper for WoC API by <b>MrZ</b>	<a href="#">GitHub</a>
Go Wrapper for bitcoin RPC by <b>Simon Ordish</b>	<a href="#">GitHub</a>

## Python

Description	Link
Python3 Wrapper for WOC API by <b>AustEcon</b>	<a href="#">GitHub</a>

## Node.js

Description	Link
JS Wrapper for WoC API by <b>baryon</b>	<a href="#">npmjs</a>
Example: How to verify merkle proofs by <b>Simon Ordish</b>	<a href="#">Gist</a>

# Transaction Processing

# TAAL Transaction Endpoints

The simplest way to broadcast transactions to BSV is to use the broadcast and the batchBroadcast endpoints. /api/v1/broadcast expects a single transaction whilst /api/v1/batchBroadcast expects a batch of them.

With both of these it is possible to send your request as text with the `application/json` mimetype or binary with the `application/octet-stream` mimetype (discussed below).

The binary format has 2 advantages:

1. The size of the request payload is half of the text equivalent.
2. Unlike with JSON, the 1st transaction in the binary stream can be processed before the entire payload has been received.

The endpoints are:

```
curl -X POST https://api.taal.com/api/v1/broadcast  
curl -X POST https://api.taal.com/api/v1/batchBroadcast
```

**Please note that these endpoints require a Taal APIKey that can be obtained for free by registering at <https://platform.taal.com>**

/api/v1/broadcast ( `application/json` )

**Request body**

```
{  
  "rawTx": "bitcoin transaction in hex..."  
}
```

**Response, if successful:**

```
201 Created  
text/plain  
<32 byte TXID in 64 hexadecimal characters>
```

## Response, if not successful

```
400 Bad request
application/json
{
  "status": number,
  "code": number,
  "error": "string"
}
```

These are the exact responses returned by the bitcoin node.

/api/v1/broadcast ( application/octet-stream )

### Request body

bitcoin transaction as a stream of bytes (binary)

### Response, if successful:

```
201 Created
text/plain
<32 byte TXID in 64 hexadecimal characters>
```

/api/v1/batchBroadcast ( application/json )

### Request body

```
[
  {
    "rawTx": "bitcoin transaction in hex..."
  },
  {
    "rawTx": "bitcoin transaction in hex..."
  },
  ...
]
```

/api/v1/batchBroadcast ( application/octet-stream )

## Request body

```
one or more bitcoin transactions as a stream of bytes (binary)
```

## Response for both text and binary requests:

```
200 OK
application/json
{
  "txs": [
    {
      "txid": "<32 byte TXID in 64 hexadecimal characters>",
      "returnResult": "success or failure",
      "resultDescription": "string"
    }
  ],
  "failureCount": number
}
```

The resultDescription contains the exact responses returned by the bitcoin node for each transaction.

---

# MAPI Endpoints

## MAPI endpoints (as per <https://github.com/bitcoin-sv-specs/brfc-merchantapi>)

```
curl -X GET https://api.taal.com/mapi/policyQuote  
curl -X GET https://api.taal.com/mapi/feeQuote  
curl -X GET https://api.taal.com/mapi/tx/{hash:[0-9a-fA-F]+}  
curl -X POST https://api.taal.com/mapi/tx  
curl -X POST https://api.taal.com/mapi/txs
```

These endpoints will point to the current live MAPI reference implementation.

It is also possible to access endpoints for a specific MAPI specification:

## MAPI 1.1 endpoints (as per <https://github.com/bitcoin-sv-specs/brfc-merchantapi/tree/v1.1>)

```
curl -X GET https://api.taal.com/mapi/v1.1/feeQuote  
curl -X GET https://api.taal.com/mapi/v1.1/tx/{hash:[0-9a-fA-F]+}  
curl -X POST https://api.taal.com/mapi/v1.1/tx  
curl -X POST https://api.taal.com/mapi/v1.1/txs
```

# Bitcoin Endpoints

## Bitcoin endpoints (only with bitcoin-enabled APIKey)

```
curl -X POST https://api.taal.com/api/v1/bitcoin
```

The possible bitcoin RPC commands that are available are:

- getbestblockhash
- gettxout
- gettxouts
- decoderawtransaction
- getrawtransaction
- getmempoolancestors
- getmempooldescendants

Examples:

```
curl \  
  --header 'Authorization: {APIKey}' \  
  --header 'Content-Type: application/json' \  
  --data-binary '{"jsonrpc": "1.0", "id":"curltest", "method": "getinfo"}' \  
  https://api.taal.com/api/v1/bitcoin  
  
curl \  
  --header 'Authorization: {APIKey}' \  
  --header 'Content-Type: application/json' \  
  --data-binary '{"jsonrpc": "1.0", "id":"curltest", "method": "getinfo"}'
```

# Electrum X Endpoints

## Electrum X endpoints (only with electrum-enabled APIKey)

```
curl -X GET https://api.taal.com/api/v1/address/{address}/info  
  
curl -X GET https://api.taal.com/api/v1/address/{address}/balance  
curl -X GET https://api.taal.com/api/v1/address/{address}/history  
curl -X GET https://api.taal.com/api/v1/address/{address}/unspent  
  
curl -X GET https://api.taal.com/api/v1/address/hash/{scriptHash}/balance  
curl -X GET https://api.taal.com/api/v1/address/hash/{scriptHash}/history  
curl -X GET https://api.taal.com/api/v1/address/hash/{scriptHash}/unspent  
  
curl -X POST https://api.taal.com/api/v1/addresses/balance  
curl -X POST https://api.taal.com/api/v1/addresses/history  
curl -X POST https://api.taal.com/api/v1/addresses/unspent
```

Examples:

```
curl \  
  --header 'Authorization: {APIKey}' \  
  https://api.taal.com/api/v1/address/1HRADRLckTpFJJskkihZp16X6jR8JVRJcr/
```

# ARC Endpoints

ARC is the successor of mAPI, the merchant API.

The URL of TAAL's ARC API is: <https://api.taal.com/arc/v1/>

For instance the policy could be fetched using the following `curl` command

```
curl --location 'https://api.taal.com/arc/v1/policy' \
--header 'Authorization: <API Key>'
```

A complete description of ARC API endpoints can be found under the following link:  
<https://bitcoin-sv.github.io/arc/api.html>.

It is recommended to use the BIP-239 transaction format for submitting transactions to ARC.

More information about ARC can be found under these links

- High-level documentation: <https://bitcoin-sv.github.io/arc/>
- Github repository: <https://github.com/bitcoin-sv/arc>
- Description of BIP-239 standard: <https://github.com/bitcoin-sv/arc/blob/master/doc/BIP-239.md>

## ARC Testnet

URL: <https://api.taal.com/arc/v1>

Authentication: Include a valid **testnet API key** in your requests.

# Transaction format and fee rate

## Fee rate

The current standard fee rate which applies to all transactions submitted to any of the TAAL transaction processing APIs is: 1 sat/kb

The minimum fee for any transaction is 1 sat.

The fee required is rounded depending on size of the transaction. The following list exemplifies the logic:

- Transactions with 1.4999... kb > size > 0.0 kb ⇒ min absolute fee required = 1 sat.
- Transactions with 2.4999... kb > size > 1.5 kb ⇒ min absolute fee required = 2 sat
- Transactions with 3.4999... kb > size > 2.5 kb ⇒ min absolute fee required = 3 sat
- ...

## Transaction format

The format of transaction is bound to the standard of BSV transactions as described [here](#).

There is a number of libraries which allow to build BSV transactions. Here is a short list:

- <https://www.npmjs.com/package/bsv>
- <https://github.com/libsv/go-bt>

### Javascript example using version 1 of bsv.js and request modules

1. Create a new Node.js project:

```
mkdir example
cd example
npm init -y
npm install bsv@1
npm install request
```

2. Save the following code to a main.js in your example folder:

```
const fs = require('fs')
const bsv = require('bsv')

// Random key produced each time...
const newKey = bsv.PrivateKey()

console.log(`Private key:      ${newKey.toString()}`)
console.log(`Mainnet address: ${newKey.toAddress('mainnet').toString()}`)

console.log(`
1. Create a funding transaction by sending 1000 satoshis (0.00001 BSV)

2. Add your Taal apiKey, and funding transaction txid and vout to 'generated.js'

3. Execute the generated code:`)

node generated.js

`)

fs.writeFileSync('generated.js', `

const bsv = require('bsv')
var request = require('request')

const privKey = bsv.PrivateKey('${newKey}')

const apiKey = ""          // Add Taal mainnet API Key
const fundingTxid = ""     // Add the funding transaction txid
const fundingVout = -1     // Replace with funding transaction vout
const returnAddress = ""   // The address to return the 1000 satoshis (1

if (apiKey === "" || fundingTxid === "" || fundingVout === -1 || returnAddress === "") {
  console.log('This generated file needs an apiKey, a fundingTxid, a fundingVout, and a returnAddress')
  process.exit(1)
}

const utxo1 = {
  txId: fundingTxid,
  outputIndex: fundingVout,
  address: privKey.toAddress('mainnet').toString(),
  script: bsv.Script.buildPublicKeyHashOut(privKey.publicKey).toHex(),
  satoshis: 1000
}

const tx1 = bsv.Transaction()
  .from(utxo1)
  .to(privKey.toAddress('mainnet').toString(), 800)`)
```

```
    .change(returnAddress)
    .sign(privKey)

const utxo2 = {
  txId: tx1.hash,
  outputIndex: 0,
  address: privKey.toAddress('mainnet').toString(),
  script: bsv.Script.buildPublicKeyHashOut(privKey.publicKey).toHex(),
  satoshis: 800
}

const tx2 = bsv.Transaction()
  .from(utxo2)
  .change(returnAddress)
  .sign(privKey)

const txBuffer = Buffer.concat([
  tx1.toBuffer(),
  tx2.toBuffer()
])

// Send the transaction to Taal API...
request.post({
  url: 'https://api.taal.com/api/v1/batchBroadcast',
  headers: {
    'Authorization': apiKey,
    'Content-Type': 'application/octet-stream'
  },
  body: txBuffer
}, function(error, response, body) {
  console.log(error, response.statusCode, body)
})
```

3. Execute the code and follow the onscreen instructions:

```
node main.js
```

# TAAL Wallet

# Introduction

TAAL Wallet is a non-custodial chrome extension for managing native BSV and tokens.

The following are the features of TAAL Wallet:

- Ability to connect a wallet to a web application.
- Manage and store your keys locally.
- Send and receive native BSV transactions
- Sign token transactions

## System Requirements

The following are the system requirements for the TAAL Wallet:

- TAAL Wallet works only with Google Chrome web browser.

# Architecture

List of tech stack used

TAAL Wallet Chrome extension is built using React and Typescript.

## Features

- [React 18](#)
- [Redux Toolkit](#)
- [BSV Javascript library](#)
- [TypeScript](#)
- [Webpack](#)
- [ESLint](#)
- [Prettier](#)
- [Chrome Extension Manifest Version 3](#)

## Chrome Extension Documentation

- [ChromeExtension](#)

## Communication

### Long lived connection

In background.js

```
chrome.runtime.onConnectExternal.addListener(port => {
  console.log('Client connected', port);
  port.onMessage.addListener(msg => {
    console.log('onExternalMessage', msg);
  });
});
```

## In web page

```
const port = chrome.runtime.connect(extensionId, { name: 'some-name' });
port.onMessage.addListener(console.log);
port.postMessage({ payload: 'anything' })
```

## In content script

```
const port = chrome.runtime.connect({ name: 'some-name' });
port.postMessage({ payload: 'anything' });
port.onMessage.addListener(msg => {
  console.log('onMessage', msg);
});
```

# One time communication (web → background.js)

## In background.js

```
chrome.runtime.onMessageExternal.addListener((payload, data, cb) => {
  console.log('onMessageExternal', { payload, data });
  if (typeof cb === 'function') {
    cb('response from background.js');
  }
  return true;
});
```

## In web page

```
chrome.runtime.sendMessage(extensionId, { payload: 'anything' }, console.
```

## In content script

```
chrome.runtime.sendMessage({ payload: 'anything' }, console.log);
```

# Terminology

## Send BSV

Ability for the user to send satoshis to a valid address on the network.

## Receive BSV

reveals a BSV address or QR code.

## History

It shows the history of transaction with transaction ID and block height.

## Add account

Ability for the user to add new account by creating new wallet with password, network selection and mnemonics .

## Satoshis

The smallest indivisible unit is a bitcoin.  $1E8$  Satoshis = 1 bitcoin

## Recovery phrase (also called seed phrase or mnemonic phrase)

A recovery phrase is a group of words, usually 12 or more that are generated when a new crypto wallet is created. You can use the recovery phrase to retrieve your crypto accounts, using any compatible wallet.

## Passphrase

The Passphrase is an advanced feature that adds a 13th word of your choosing to your recovery phrase.

Using a Passphrase will cause an entirely different set of addresses to be created which cannot be accessed via the 12-word recovery phrase alone.

Aside of adding another layer, the Passphrase grants you plausible deniability when under duress.

If using a Passphrase, it's key to store it securely and remember it perfectly, character for character.

### **Get Balance**

Enables user to get the latest balance reflected in the account.

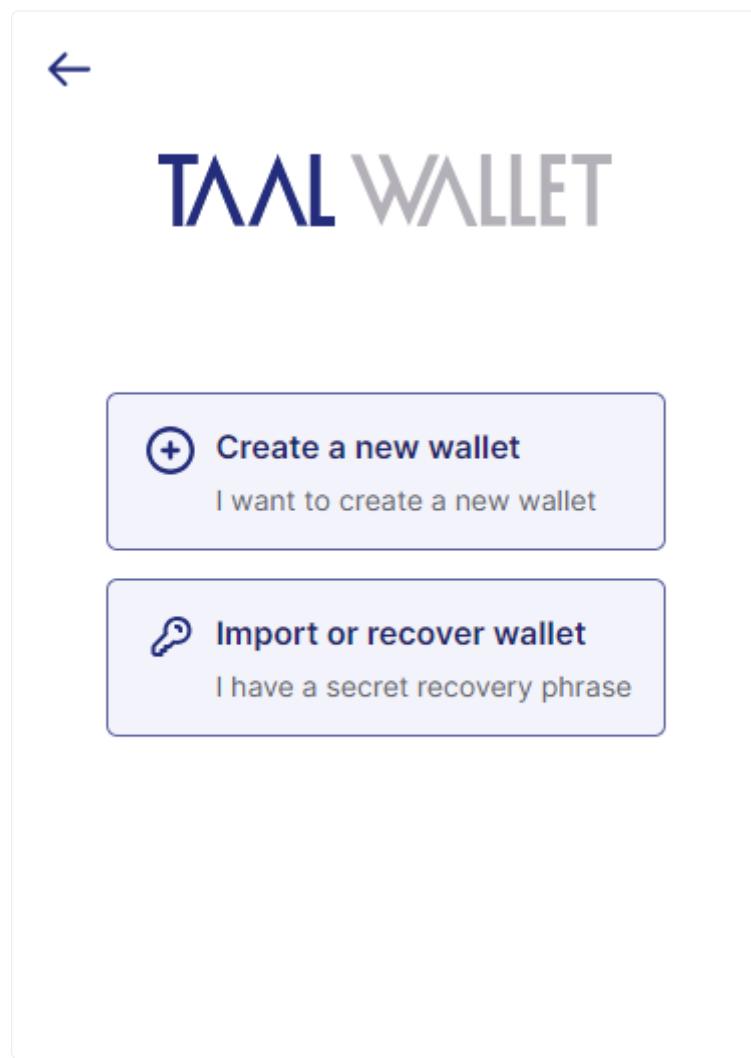
### **Airdrop**

Testnet only. Will send a small amount of BSV test coins to your account.

# UI Elements

## Welcome to TAAL Wallet window

This is the welcome window of the TAAL Wallet.



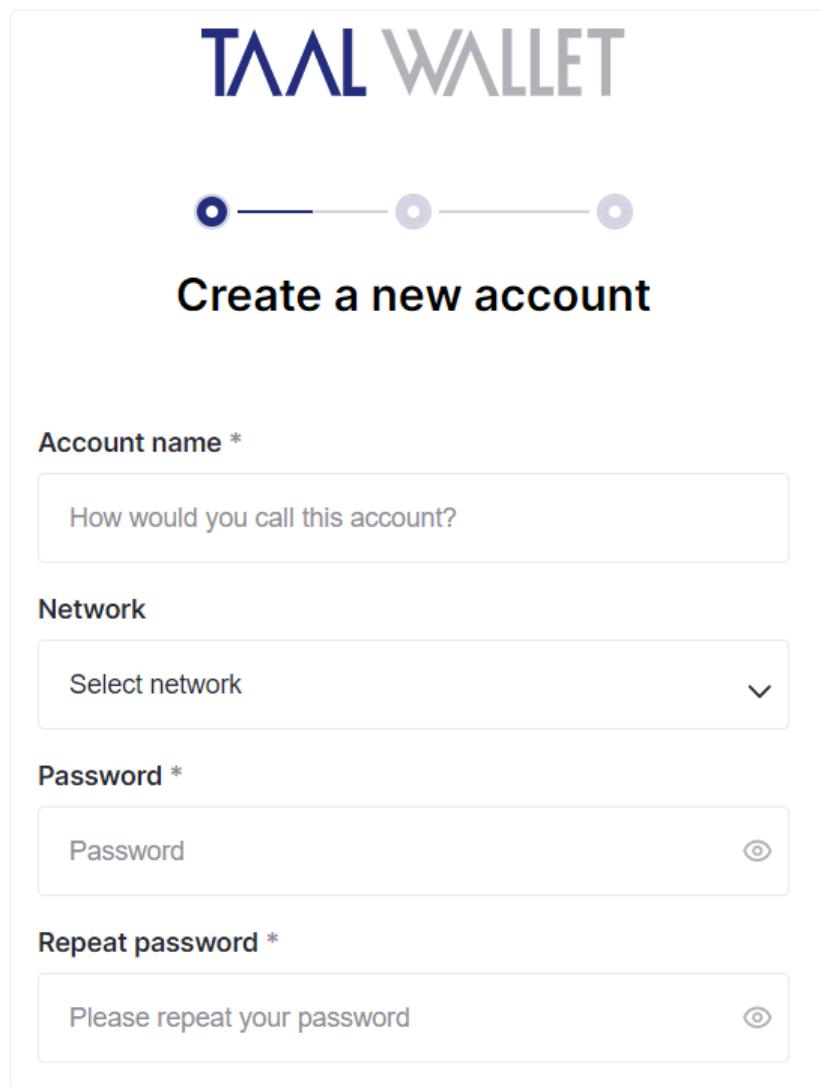
The following table describes the UI elements available on this window.

UI Elements	Description
CREATE A WALLET button	Displays the <i>Create a new Wallet</i> window, where you can create a new wallet.
IMPORT WALLET button	Displays the <i>Import your Wallet</i> window, where you can import an existing wallet if you have the secret recovery phrase.

## Create a new wallet window

You can use this window to create a new wallet.

**To reach this window:** On the *Welcome to TAAL Wallet* window, click **CREATE A WALLET** or on the *Wallet* window, click on **Add new account**.



The image shows the 'Create a new account' window of the TAAL Wallet application. At the top, the TAAL Wallet logo is displayed. Below it, there is a progress bar consisting of three circles connected by lines. The first circle is filled blue, while the other two are grey. The text 'Create a new account' is centered below the progress bar. The form fields include: 'Account name \*' with a placeholder 'How would you call this account?'; 'Network' with a dropdown menu labeled 'Select network'; 'Password \*' with a password input field; and 'Repeat password \*' with a password input field. Each password field has an eye icon to the right for visibility.

I have read and agreed to the [Terms of use](#).

[Back](#)

[Proceed](#)

Step 1



## Back up your secret phrase



### Keep it safe

Write down your secret phrase and keep it in a safe place



### Don't share it

Never share your secret phrase with anyone or, they will have access to your wallet.

[Back](#)

[Proceed](#)



## Your secret phrase

spread rural zoo vivid omit cattle mango parade wre  
ck tumble aim fix

 Copy to clipboard[Back](#)[Proceed](#)

# TAAL WALLET



Please select first 3 words of your secret phrase in the correct order

zoo

mango

vivid

tumble

wreck

fix

parade

spread

rural

cattle

omit

aim

[Back](#)

The following table describes the UI elements available on this window.

UI Elements	Description
Account name field	The name of the account. This field is mandatory.
Password field	The 8 characters password for the account. This field is mandatory.
Repeat password field	The password for the account same as provided in the previous field. This field is mandatory.
Network drop-down list	The available networks, and the options are: <ul style="list-style-type: none"> <li>● Testnet</li> <li>● Mainnet</li> </ul> This field is mandatory.
Recovery phrase	It is a group of words, usually 12 or more that are generated when a new crypto wallet is created. <b>Note:</b> Here is your 12 words recovery phrase. It is incredibly important to take a note of this and keep it safe as without you may lose access to your wallet and the funds contained within.

## Import your Wallet window

You can use this window to import an existing wallet.

**To reach this window:** On the *Welcome to TAAL Wallet* window, click **IMPORT WALLET** or on the *Wallet* window, click > **Import account**.



**CHOOSE WALLET TYPE****Standard Security**

No passphrase

**Passphrase Security**

Passphrase is required

**TAAL WALLET****Recover or Import Standard Wallet****Account name \***

How would you call this account?

**Network**

Select network

**Enter your recovery phrase**

```
word #1 word #2 word #3 word #4 word #5 word #6  
word #7 word #8 word #9 word #10 word #11 word  
#12
```

Paste or type your phrase in the right sequence

**Back****Proceed**

Recover an account secured with a standard recovery phrase

**TAAL WALLET**



## Recover or Import Hidden Wallet

Account name \*

How would you call this account?

Network

Select network



Enter your recovery phrase

word #1 word #2 word #3 word #4 word #5 word #6  
word #7 word #8 word #9 word #10 word #11 word  
#12

Paste or type your phrase in the right sequence

Passphrase \*

Passphrase



Back

Proceed

Recover an account secured with a standard recovery phrase and a passphrase

The following table describes the UI elements available on this window.

UI Elements	Description
create a new one button	Displays the <i>Create a new Wallet</i> window, where you can create a new wallet.
Account name field	The name of an existing account. This field is mandatory.
Password field	The 8 characters password for the account. This field is mandatory.
Repeat password field	The password for the account same as provided in the previous field. This field is mandatory.
Network drop-down list	The available networks, and the options are: <ul style="list-style-type: none"><li>● Testnet</li><li>● Mainnet</li></ul> This field is mandatory.
Recovery phrase field	It is a group of words, usually 12 or more that are generated when a new crypto wallet is created.
Passphrase	The Passphrase is an advanced feature that adds a 13th word of your choosing to your recovery phrase.

## Wallet main window

The main window of the TAAL Wallet.

The following table describes the UI elements available on this window.

UI Elements	Description
button	Displays the wallets and selected network.
Add new account button	Displays the <i>Create a new Wallet</i> window, where you can create a new wallet.
Import account button	Displays the <i>Import your Wallet</i> window, where you can import an existing wallet.
button	Displays all the menu options available.
Home button	Displays <i>Home</i> window.
Select Wallet button	Displays the <i>Your Wallets</i> window, where you can select the Wallet you want.
Send BSV button	Displays the <i>Send BSV</i> window, where you can send satoshis to a valid address on the network.
Receive BSV button	Displays the <i>Receive BSV</i> window, where you can view the BSV address or QR code of the account.
History button	Displays the <i>Your Wallet's history</i> window, where you can view the history of transaction with transaction ID and block height.
Tokens button	Displays the <i>Your Tokens</i> window, where you can view the tokens.
Options button	Display the Access Control List window, where you can view and delete the Websites with access to TAAL Wallet.
Airdrop button	Testnet only. Will send a small amount of BSV test coins to your account.
Lock Wallet button	Locks the wallet.

## Home window

This window displays the current account, wallets details, and balance.

**To reach this window:** on the *Wallet* main window, click > **Home**.

The following table describes the UI elements available on this window.

UI Elements	Description
Current account	The name of the current account.
	Edits the name of the current account.
Name field	The name of the wallet.
Path field	The path of the wallet
Address field	The address of the wallet.
Balance field	The balance in satoshis available in the wallet.
Updated at field	The date and time when the wallet was last updated.
SELECT ANOTHER WALLET button	Displays the <i>Your Wallets</i> window, where you can view the wallets available in your current account, and you can select the wallet you want.
Balance	The balance in satoshis available in the current account.
Refresh balance or refresh button	Refreshes the balance.
SEND BSV button	Displays the <i>Send BSV</i> window, where you can send satoshis to a valid address on the network.
RECEIVE BSV button	Displays the <i>Receive BSV</i> window, where you can view the BSV address or QR code of the account.
HISTORY button	Displays the <i>Your Wallet's history</i> window, where you can view the history of transaction with transaction ID and block height.
TOKENS button	Displays the <i>Your Tokens</i> window, where you can view the tokens.

## Your Wallets window

This window displays the wallets available in your current account, and you can select the wallet you want.

**To reach this window:** On the *Wallet* main window, click > **Select Wallet**, or on the *Home* window, click **SELECT ANOTHER WALLET**.

The following table describes the UI elements available on this window.

UI Elements	Description
CREATE A NEW WALLET button	Display the <i>Create a New Wallet</i> window, where you can create a new wallet.
Wallet section	Display the wallet details, such as Name, address, Path, Balance, and Updated at.
SELECT button	Enables you to select the wallet you want.

## Send BSV window

You can use this window to send satoshis to a valid address on the network.

**To reach this window:** On the *Wallet* main window, click > **Send BSV** or on the *Home* window, click **SEND BSV**.

The following table describes the UI elements available on this window.

UI Elements	Description
Destination address field	The address to which you want to send satoshis. This field is mandatory.
Amount in satoshis (max 1002500) field	The amount in satoshis you want to send. This field is mandatory.
CANCEL button	Cancels the transaction.
SEND button	Sends satoshis to a valid address on the network.

# Receive BSV

This window displays the BSV address or QR code of the account.

-  It works only with BSV. If you send any other cryptos, it will result in the loss of funds.

**To reach this window:** On the *Wallet* main window, click > **Receive BSV** or on the *Home* window, click **RECEIVE BSV**.

The following table describes the UI elements available on this window.

UI Elements	Description
Your BSV address: field	The BSV address of the account.
Click to copy button	Copies the BSV address of the account.
QR code	The QR code of the account.

# Your Wallet's history window

This window displays the history of transaction with transaction ID and block height.

**To reach this window:** On the *Wallet* main window, click > **History** or on the *Home* window, click **HISTORY**.

The following table describes the UI elements available on this window.

UI Elements	Description
Refetch history button	Retrieves the history of the wallets.
TX ID field	The ID of the transaction.

## Your token window

This view displays the tokens.

**To reach this window:** On the *Wallet* main window, click > **Tokens** or on the *Home* window, click **TOKENS**.

## Access Control List window

This window displays the Websites with access to TAAL Wallet, and you can also delete them.

**To reach this window:** On the *Wallet* main window, click > **Options**.

# Tutorial

## Install TAAL Wallet from Chrome Web Store

The following are the steps to install TAAL Wallet in your system from Chrome Web Store:

1. Go to [Chrome Web Store](#).  
The *chrome web store* window is displayed.
2. In the **Search the store** field, search for TAAL Web3 Wallet.
3. Click **Add to Chrome**.  
You can see the TAAL Wallet in your extensions.
4. Click TAAL Wallet.  
The *Terms of Service agreement* window is displayed.
5. Click **ACCEPT**.  
The *Welcome to TAAL Wallet* window is displayed and you are ready to use the TAAL Wallet.



TAAL Wallet works only with Google Chrome web browser.

## Install TAAL Wallet using the code from GitHub

The following are the steps to install TAAL Wallet using the code from GitHub:

1. Clone the code available at: <https://github.com/TAAL-GmbH/taal-wallet>.
2. In your **Command Prompt**, go to the working directory.
3. Run `yarn`.
4. Run `yarn dev`.
5. Go to the Google Chrome web browser.

6. Click **Extensions > Manage Extensions**.  
The *Extensions* window is displayed.
7. In the *Extensions* window, enable **Developer mode**.
8. Click **Load unpacked**.  
The **Select the extension directory**. dialog box is displayed.
9. In the **Select the extension directory**. dialog box, select the **dist** folder from the project.  
The TAAL Wallet extension is available in your **Extensions**.
10. Click TAAL Wallet.  
The *Terms of Service agreement* window is displayed.
11. Click **ACCEPT**.  
The *Welcome to TAAL Wallet* window is displayed and you are ready to use the TAAL Wallet.

## Customise the password length

Copy `sample.env` to `.env` and customise it with your requirements, Eg.

`PASSWORD_MIN_LENGTH=1`

## Create a new Wallet

The following are the steps to create a wallet:

1. On the *Welcome to TAAL Wallet* window, click **CREATE A WALLET** or on the *Wallet* window, click **> Add new account**.  
The *Create a new wallet* window is displayed.
2. In the **Account name** field, enter the account name.
3. In the **Password** field, enter the 8 characters password.
4. In the **Repeat password** field, repeat the password.
5. In the **Network** list, select the network, such as Taalnet, Testnet, and Mainnet.
6. In the **Secret phase** field, you are provided with the 12 words seen phrase.
7. Click **NEXT**.  
A new wallet is created.

**TAAL Wallet**

Back

 Create a new wallet

or [import existing one](#)

Account name \*

 X

Password \*

 • X

Repeat password \*

 • X

Network \*

 ▼

Secret phrase \*

payment stem transfer point clip rival eye inmate  
hour inform absent frame

 The secret phrase is your 12 words seen phrase, it is important to keep it safe and without it you may lose access to your wallet and the funds in it.

**NEXT**

## Import an existing Wallet

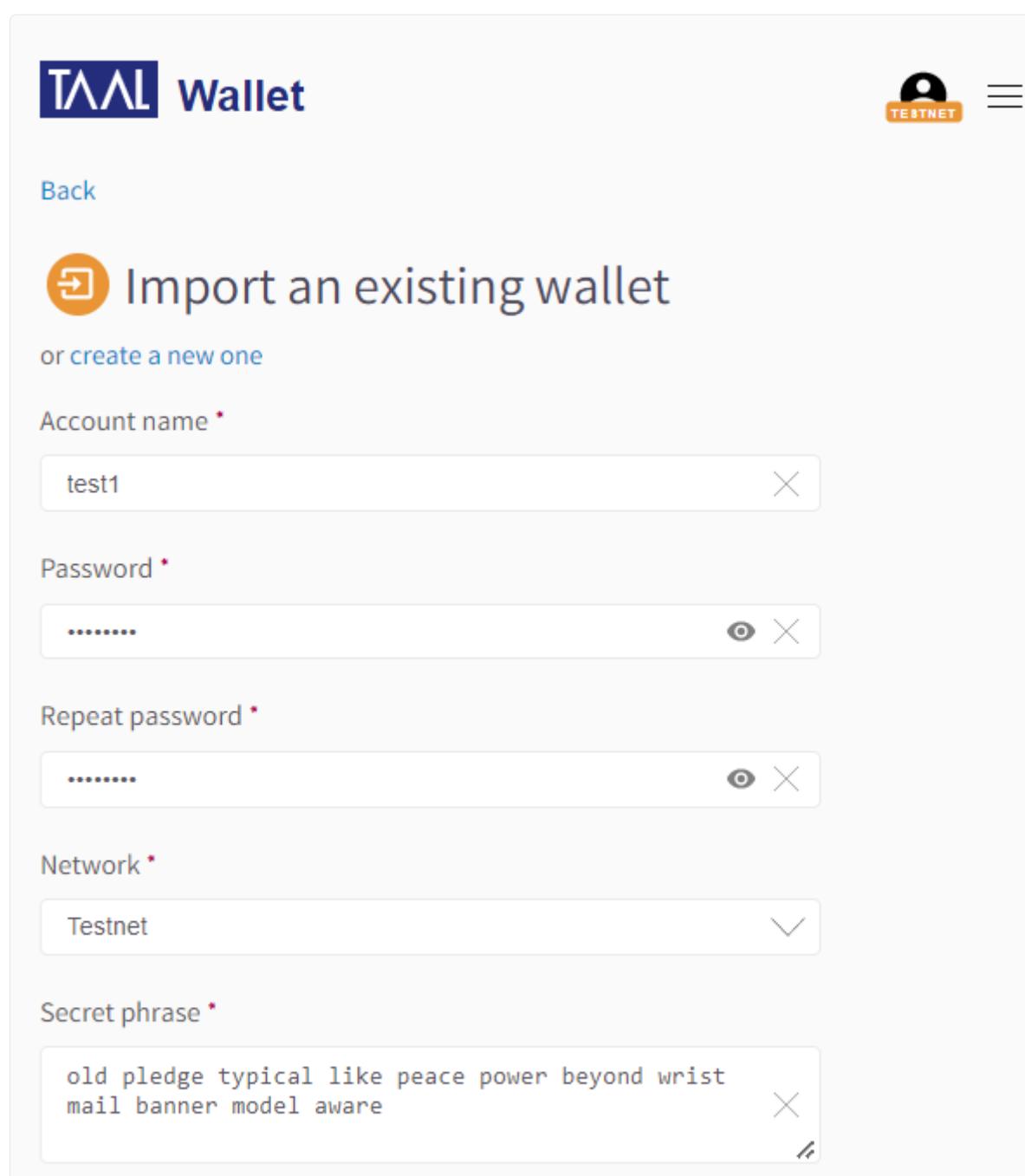
The following are the steps to import an existing wallet:

1. On the *Welcome to TAAL Wallet* window, click **IMPORT WALLET** or on the *Wallet* window, click > **Import account**.

The *Import your Wallet* window is displayed.

2. In the **Account name** field, enter the name of an existing account.
3. In the **Password** field, enter the 8 characters password.
4. In the **Repeat password** field, repeat the password.
5. In the **Network** list, select the network, such as Taalnet, Testnet, and Mainnet.
6. In the **Secret phrase** field, enter the 12 words seen phrase provided when you created the this account.
7. Click **NEXT**.

An existing wallet is imported.



The screenshot shows the 'Import an existing wallet' page of the Taal Wallet application. At the top, there's a header with the 'Taal Wallet' logo, a 'TESTNET' button, and a menu icon. Below the header, there's a 'Back' link. The main title is 'Import an existing wallet' with an orange circular icon containing a keyhole symbol to its left. Below the title, there's a sub-instruction 'or [create a new one](#)'. The form fields are as follows:

- Account name \***: A text input field containing 'test1'.
- Password \***: A password input field showing '.....'.
- Repeat password \***: A password input field showing '.....'.
- Network \***: A dropdown menu currently set to 'Testnet'.
- Secret phrase \***: A text input field containing the 12-word phrase: 'old pledge typical like peace power beyond wrist mail banner model aware'.

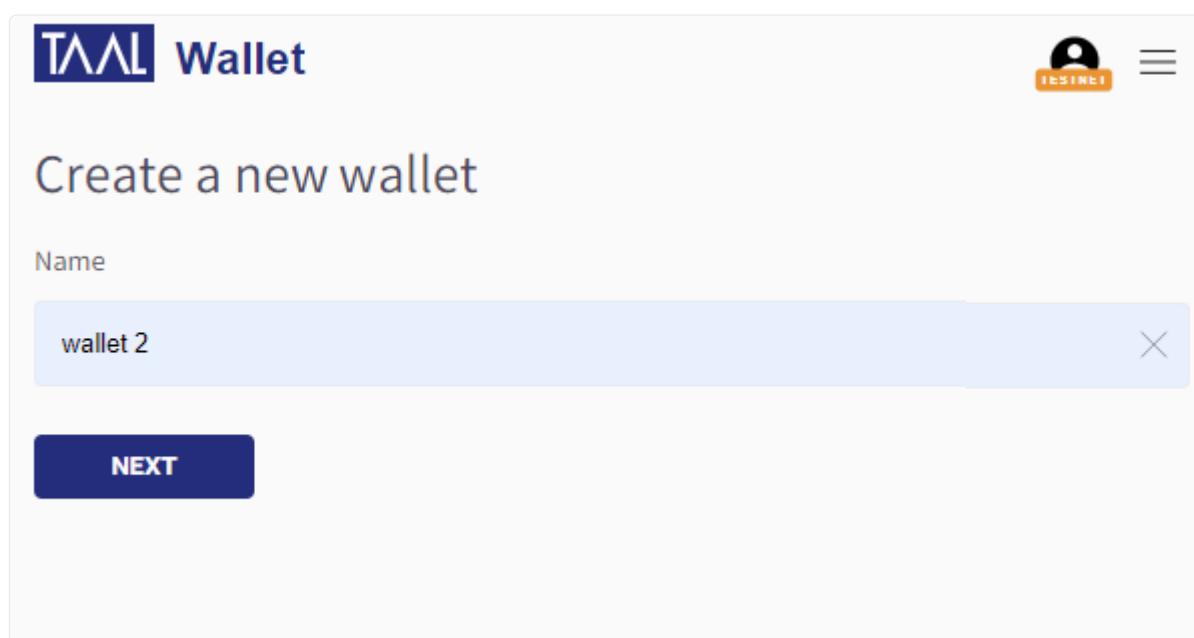
**NEXT**

## Create a new Wallet in the current account

The following are the steps to create a new wallet in the current account:

1. On the **Wallet** main window, click > **Select Wallet**.  
The *Your Wallets* window is displayed.
2. On the *Your Wallets* window, click **CREATE A NEW WALLET**.  
The *Create a New Wallet* window is displayed.
3. In the **Name** field, enter the name of the wallet.
4. Click **NEXT**.

A new wallet is created



## Send BSV

The following are the steps to send BSV:

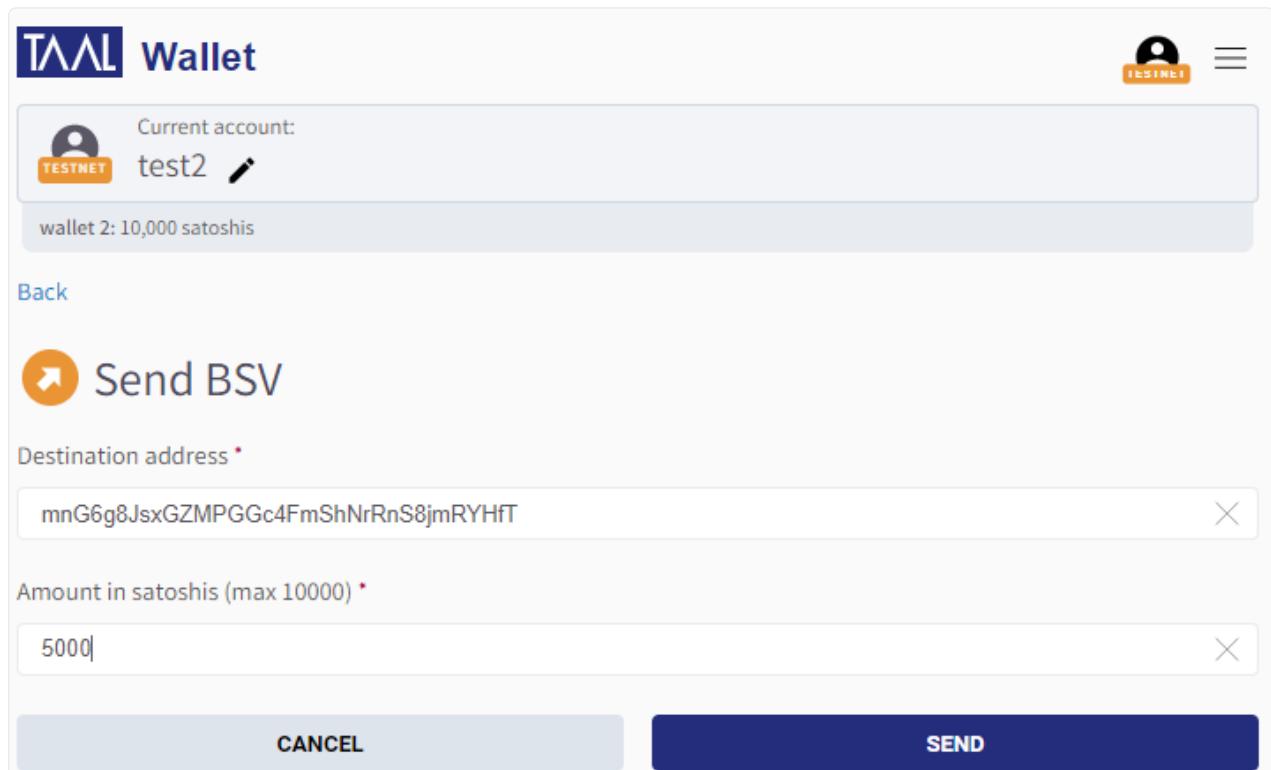
1. On the *Wallet* main window, click > **Send BSV** or on the *Home* window, click

## SEND BSV.

The *Send BSV* window is displayed.

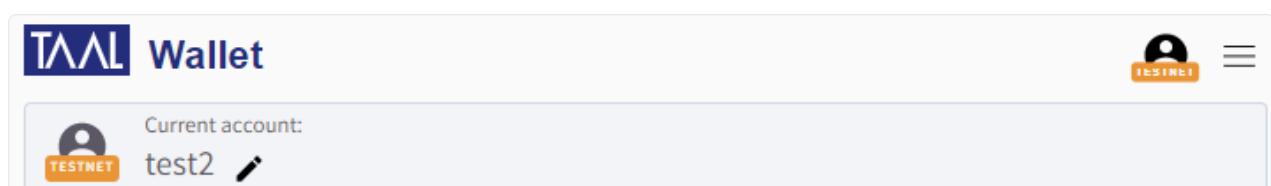
2. In the **Destination address** field, enter the address to which you want to send satoshis.
3. In the **Amount in satoshis (max 1002500)** field, enter the amount in satoshis you want to send.
4. Click **SEND**.

The satoshis is sent to a valid address on the network.



## Receive BSV

1. On the *Wallet* main window, click > **Receive BSV** or on the *Home* window, click **RECEIVE BSV**.  
The *Receive BSV* window is displayed.
2. Click to copy the BSV address of the account or scan the QR code.



Wallet-0: 0 satoshis

[Back](#)

## Receive BSV

Your BSV address:

 moAA2zL9BHzYtU6etmWibeChTZ3PYKjm5b



 It works only with BSV. If you send any other cryptos, it will result in the loss of funds.

## View History

**To view History:** On the *Wallet* main window, click > **History** or on the *Home* window, click **HISTORY**.

The *Your Wallet's history* window is displayed in which you can view the history of transaction with transaction ID and block height.

 **Wallet**  

Current account:  
test2 

Wallet-0: 0 satoshis

[Back](#)

## Your wallet's history

TX ID: [e612195f436c02b8e2f13889a2ad12934811d59d0e83615fdf2706c761bc4c7e](#)

# Airdrop

**To airdrop:** On the *Wallet* main window, click > **Airdrop**.

**Testnet** only. Will send a small amount of BSV test coins to your account.

# 1Sat Ordinals tokens API

# Introduction

1SatOrdinals in an implementation of Ordinals running on the BSV blockchain.

See the  [Protocol Specification](#) | [1Sat Ordinals](#) for more information on 1Sat Ordinals and the [Ordinals Docs](#) for more information on Ordinals and ordinal theory.

# Terminology

## Satoshis

The smallest indivisible unit of a bitcoin.  $1E8$  Satoshis = 1 bitcoin

## Script

Low level bitcoin script.

## SDK

Software Developer Kit is a software library to make certain tasks easier for a software developer.

## UTXO

An Unspent Transaction Output is an amount of bitcoin that can be spent by the recipient.

## Inscription ID

An outpoint representing a particular inscription, comprised of a transaction ID and output index with the following formatting: `txid\_vout`

## Inscription Number

This is an auto-incrementing id given to each inscription as they appear in mined blocks, starting with inscription #0. Because inscription numbers are dependent on the order of transactions in a block, inscription numbers are not assigned until a transaction is confirmed. This also makes inscription number sensitive to reorgs.

## Origin

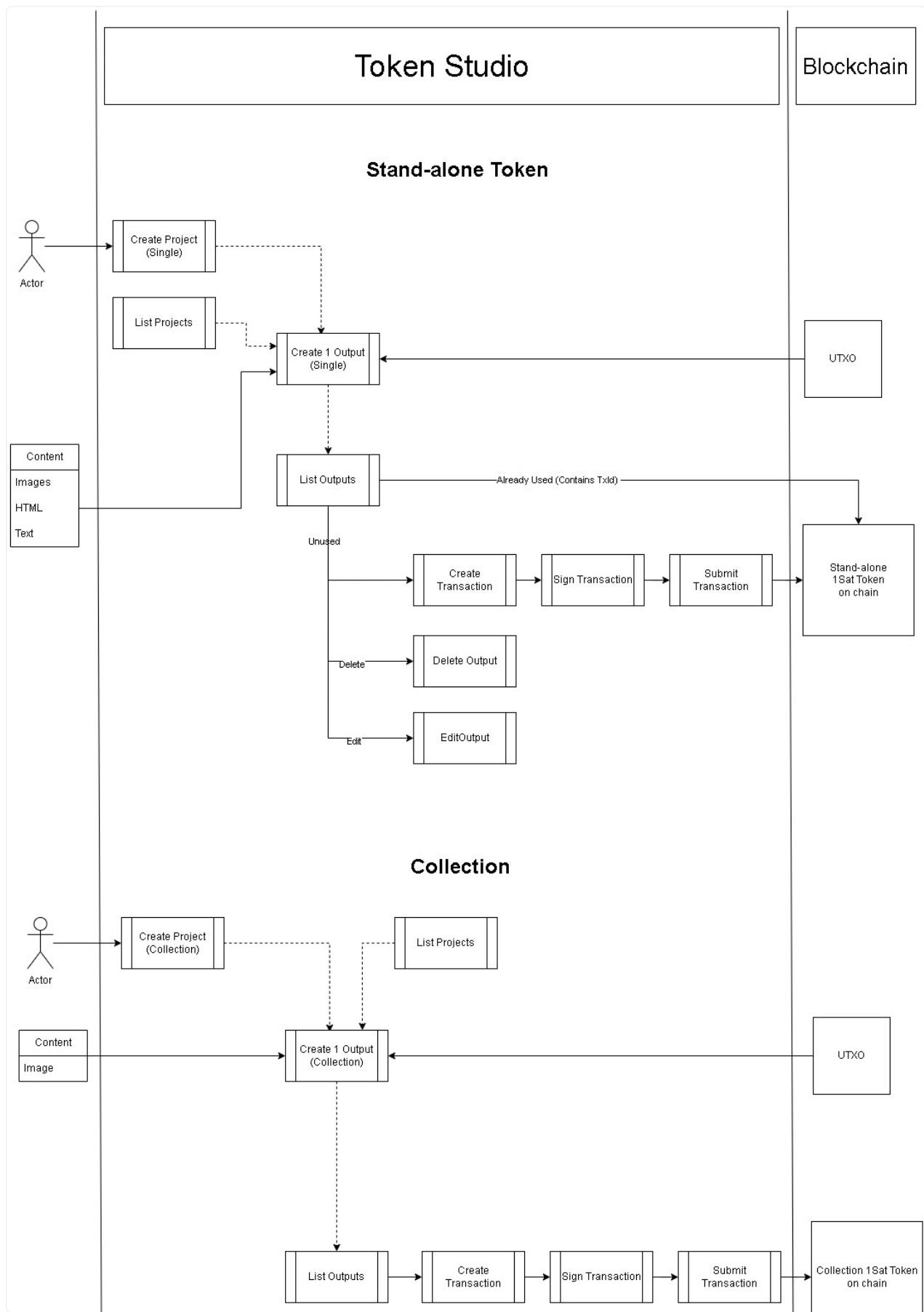
This is like a "low resolution" ordinal number. It represents the last time a particular ordinal became a 1 satoshi output. Since utxos can be split and joined over time, an ordinal can have more than one origin. This is a 1Sat protocol specific concept, not

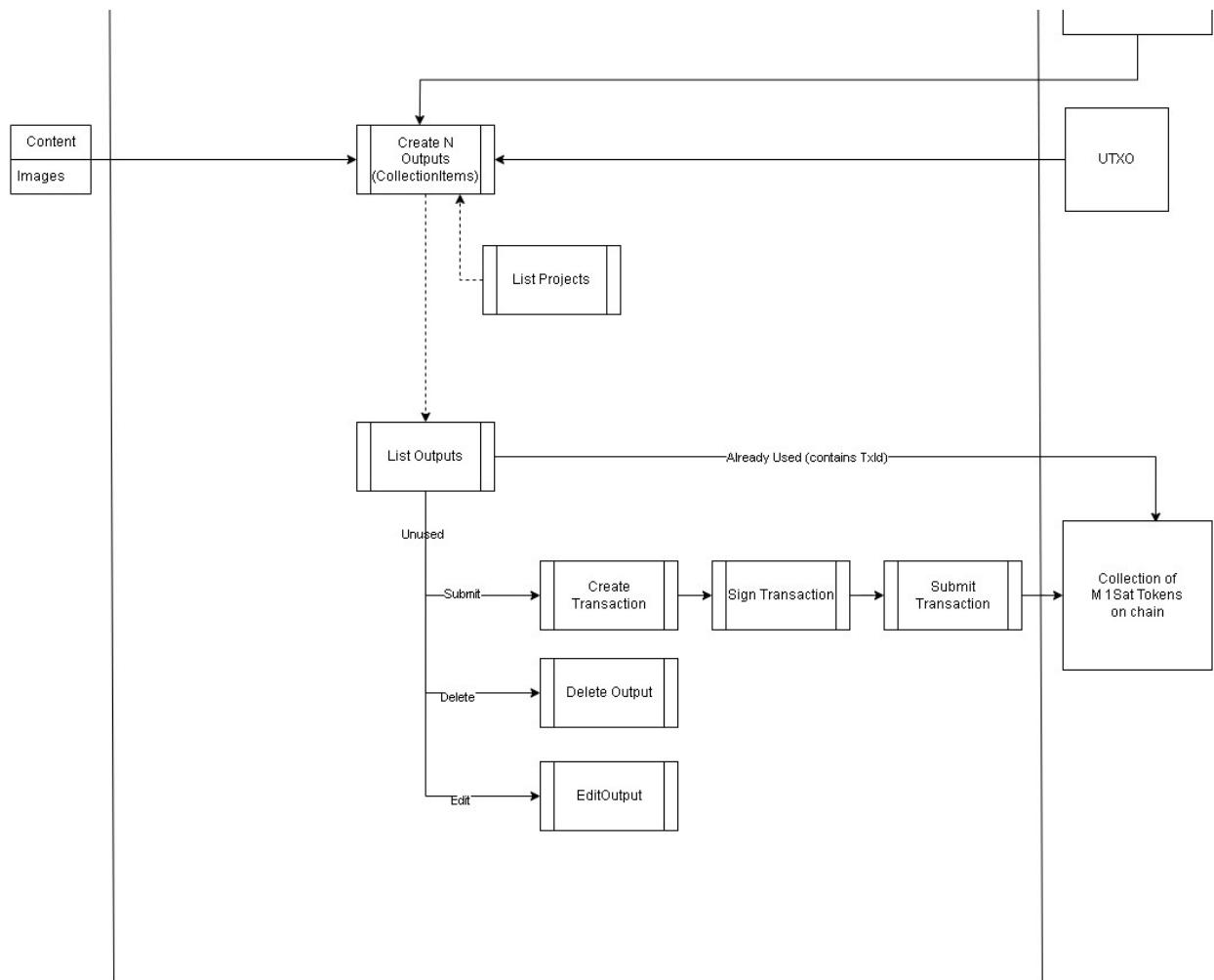
present in the original ordinals protocol.

## Ordinal Number

An ordinal is an identifier given to a specific satoshi. It is given when the sat comes into existence during a coinbase transaction. To learn more about this concept read the original "ordinal theory" documentation.

# Flow Diagram





# Basic Tutorial - Node

## Github Repository

<https://github.com/TAAL-GmbH/1sat-tutorial>

```
mkdir 1sat-tutorial
```

```
cd 1sat-tutorial
```

Demo code using our 1Sat Token Studio API can be found on Github so you can clone our repository or if you want to do it manually, we need to make now a node project

```
npm init
```

and answer in the questions.

Let's add the dependencies

```
npm install bsv@1.5.4  
npm install axios
```

Make sure your package.json looks like this (you might need to change the type to "module" and add a new entry to run the tutorial- "tutorial": "node index.js"

```
{
  "name": "1sat-tutorial",
  "version": "1.0.0",
  "description": "1sat-tutorial",
  "type": "module",
  "main": "index.js",
  "scripts": { "test": "echo \\\"Error: no test specified\\\" && exit 1",
    "tutorial": "node index.js"
  },
  "author": "TAAL",
  "license": "ISC",
  "dependencies": {
    "dependencies": { "axios": "^1.7.2",
      "bsv": "^1.5.4"
    }
  }
}
```

Now we are ready to start coding. Create the main file you specified during the setup e.g. `index.js` in the root of the `1sat-tutorial` folder as well as the helping file `1sat_utils.cjs` using your editor or IDE of choice.

1. Copy the code from our files and paste.
2. in `index.js` you will need to replace the sample API key from your TAAL API Key
3. You can also tune which steps to run, only the creation of a single/standalone token, the inscription of a collection with one item or individual steps you may want to debug in depth e.g.

```
const STEPS_TO_TEST =[ 'single', 'collection'];
//const STEPS_TO_TEST =['1.1','1.2','1.3','1.4','1.5', '1.6', '1.7'];
//const STEPS_TO_TEST =['2.1','2.2','2.3','2.4','2.5', '2.6', '2.7',
'2.8', '2.9'];
```

You can test now. If all was properly configured you should see something like this:

```
> 1sat-tutorial@1.0.0 tutorial
> node index.js
Step 1.0 - Setup Standalone Project
Public Key:02509aa4fb3fees371a086d628d5f012ee5ce5b31073c95478a7f3dc21dc6b0a56
Address :mpcCvaQHwBAMaNxqk66ffwjm5FEWUzu7
Step 1.1 - Create STANDALONE Project
{"name":"1satcol project 1716461684783","isFungible":true,"type":"single","tokenProtocol":"OneSat0
dinal"}
https://console.test.taal.com/token-studio/api/v1/project/create
true
Step 1.2 - get Project Details
https://console.test.taal.com/token-studio/api/v1/project/87948b35-9b61-43f1-b99a-f68547d9edc6
true
Step 1.3 - get list of projects
https://console.test.taal.com/token-studio/api/v1/project/
```

```
true
Step 1.4 - create a single output for a standalone token
https://console.test.taal.com/token-studio/api/v1/token/one-sat-ord/create-output
true
Step 1.5 - get list of project outputs
https://console.test.taal.com/token-studio/api/v1/project/87948b35-9b61-43f1-b99a-f68547d9edc6/output-list
true
Step 1.6 - Create Transaction
("projectId": "87948b35-9b61-43f1-b99a-f68547d9edc6", "publicKey": "02509aa4f83fee3371a006d628d5f012e5ce5b31073c95478a7f3dc21dc6b0a56", "dstAddress": "mppcvQHwSBAMnXqK6ffWfjm5FEUzu7", "outputList": [{"outputIndex": 0, "txId": "b07b5417924eb28c6e77ae500000017efac1b39d63a08881be10fd13dc4984"}])
https://console.test.taal.com/token-studio/api/v1/token/one-sat-ord/create-transaction
false
true
Step 1.7 - Submit Transaction
https://console.test.taal.com/token-studio/api/v1/submit
true
```

```
Step 2.1 - Create COLLECTION Project
("name": "1satcol project 1716461688098", "isFungible": true, "type": "collection", "tokenProtocol": "OneSatOrdinal")
https://console.test.taal.com/token-studio/api/v1/project/create
true
Step 2.2 - create a collection output
https://console.test.taal.com/token-studio/api/v1/token/one-sat-ord/create-output
true
Step 2.3 - get list of project outputs
https://console.test.taal.com/token-studio/api/v1/project/f33fc091-514e-45e8-a9a8-960645757a9b/output-list
true
true
Step 2.4 - Create Transaction
("projectId": "f33fc091-514e-45e8-a9a8-960645757a9b", "publicKey": "02509aa4f83fee3371a006d628d5f012e5ce5b31073c95478a7f3dc21dc6b0a56", "dstAddress": "mppcvQHwSBAMnXqK6ffWfjm5FEUzu7", "outputList": [{"outputIndex": 0, "txId": "200e726dd1787ae3c700e034a95abb3ad603a27ec3c9d5e1dd60d98b2437d89"}])
https://console.test.taal.com/token-studio/api/v1/token/one-sat-ord/create-transaction
41e61c00-81c2-42f6-9527-fce16850f138
true
Step 2.5 - Submit Transaction (Collection)
https://console.test.taal.com/token-studio/api/v1/submit
true
Step 2.6 - create collectionItem output
https://console.test.taal.com/token-studio/api/v1/token/one-sat-ord/create-output
true
true
Step 2.7 - get list of project outputs
https://console.test.taal.com/token-studio/api/v1/project/f33fc091-514e-45e8-a9a8-960645757a9b/output-list
true
true
Step 2.8 - Create Transaction
("projectId": "f33fc091-514e-45e8-a9a8-960645757a9b", "publicKey": "02509aa4f83fee3371a006d628d5f012e5ce5b31073c95478a7f3dc21dc6b0a56", "dstAddress": "mppcvQHwSBAMnXqK6ffWfjm5FEUzu7", "outputList": [{"outputIndex": 0, "txId": "c0642792-42c7-463a-9f9e-e0977544a309"}])
https://console.test.taal.com/token-studio/api/v1/token/one-sat-ord/create-transaction
100e2011-94c5-4bc8-9d33-cdd40b7ec36
true
Step 2.9 - Submit Transaction (CollectionItem)
https://console.test.taal.com/token-studio/api/v1/submit
true
3614ef7782428bce7c5d367796a04adbe80018ec584dfaff7c67e205be0d
```

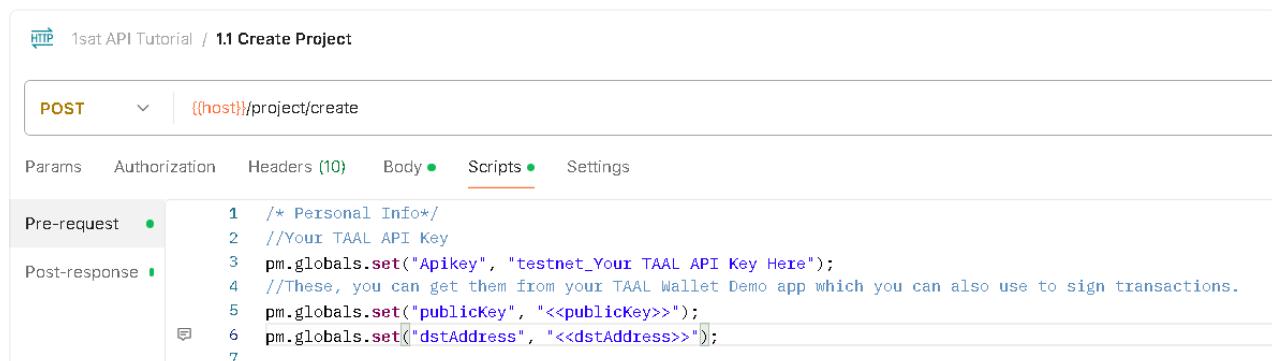
## Add label

# Basic Tutorial - Postman

You can download our Postman Collection from here:

<https://github.com/TAAL-GmbH/1sat-tutorial>

Once imported you will need to type in your personal values in the 1.1 Create Project Pre-request script

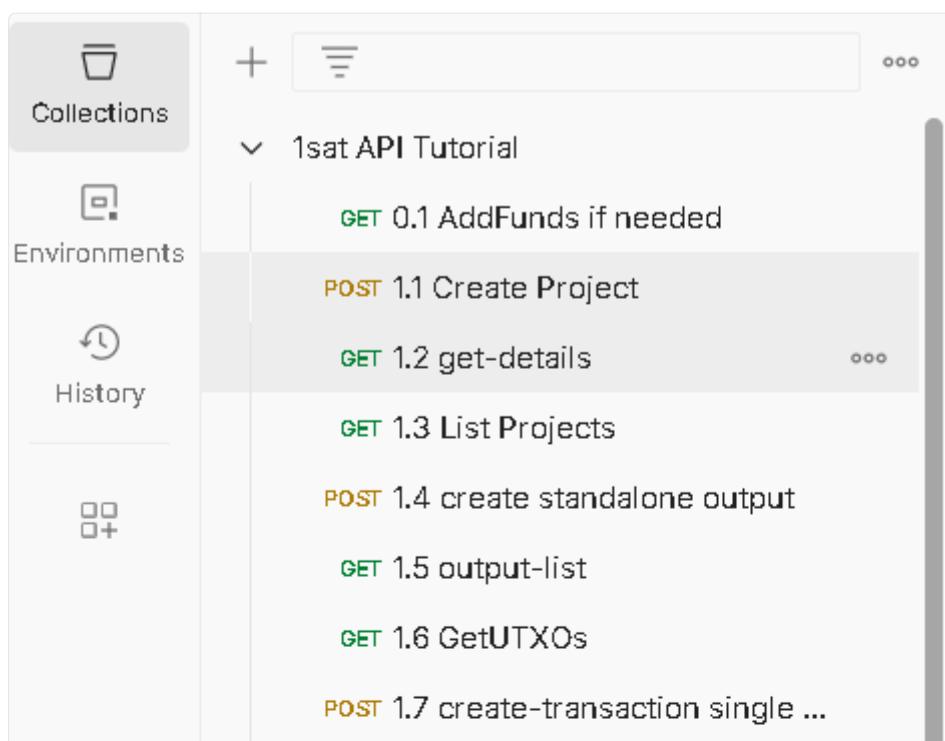


```
/* Personal Info*/
//Your TAAL API Key
pm.globals.set("Apikey", "testnet_Your TAAL API Key Here");
//These, you can get them from your TAAL Wallet Demo app which you can also use to sign transactions.
pm.globals.set("publicKey", "<>publicKey<>");
pm.globals.set("dstAddress", "<>dstAddress<>");
```

Once done, you can go through the steps one by one to create:

1.- A Standalone token (HTML token)

2.- A Collection (Image Token) with one collectionItem (Text token)



**POST** 1.8 submit tx single token

**GET** 1.9 GetTransaction Info

**POST** 2.1 Create Collection Project

**POST** 2.2 create collection output

**GET** 2.3 output-list

**GET** 2.4 GetUTXOs

**POST** 2.5 create-transaction collec...

**POST** 2.6 submit tx collection token

**POST** 2.7 create collectionItem out...

**GET** 2.8 output-list

**GET** 2.9 GetUTXOs

**POST** 2.10 create-transaction colle...

**POST** 2.11 submit tx collectionItem ...

# API

This API will allow you to create 1SatOrdinals Single and Collection Tokens.

## Common

An Api Key must be passed in the header of all requests see [Authentication](#)

## Create a Token Studio Project for your Standalone - Collection

HTTP Request

```
POST https://platform.taal.com/token-studio/api/v1/project/create
```

cURL

```
curl --location 'https://platform.taal.com/token-studio/api/v1/project/create' \
--header 'Apikey: ## VALID_API_KEY ##' \
--header 'Content-Type: application/json' \
--data '{
  "name": "1satcol project 1718100516565",
  "isFungible": false,
  "type": "single", //collection
  "tokenProtocol": "OneSatOrdinal"
}'
```

Response JSON structure

```
{
  "success": boolean,
  "data": {
    "projectUid": <uid>
  }
}
```

# Get a list of your Token Studio projects

It will return all projects associated with your API Key.

HTTP Request

```
GET https://platform.taal.com/token-studio/api/v1/project/
```

cURL

```
curl --location 'https://platform.taal.com/token-studio/api/v1/project/'  
--header 'Apikey: ## VALID_API_KEY ##'
```

Response JSON structure

```
{  
    "success": boolean,  
    "data": {  
        "projectList": [  
            {  
                "uid": <uid>,  
                "name": string,  
                "network": string,  
                "offChainData": {  
                    "isFungible": boolean  
                },  
                "tokenProtocol": string  
            }  
        ]  
    }  
}
```

# Get details for a specific project id

Returns a project for a valid project id.

HTTP Request

```
GET https://platform.taal.com/token-studio/api/v1/project/<projectId>
```

## cURL

```
curl --location 'https://platform.taal.com/token-studio/api/v1/project/'  
--header "Apikey: ## VALID_API_KEY ##"
```

## Response JSON structure

```
{  
    "success": true,  
    "data": {  
        "uid": <uid>,  
        "name": string,  
        "tokenProtocol": string,  
        "network": string,  
        "isFungible": boolean,  
        "type": string  
    }  
}
```

**URL Parameters**

Parameter	Description
projectId	A valid project Id

## Output creation

Creates an output (single, collection, collectinItem) in the project.

Single projects can only contain single/standalone outputs

Collection projects need first a collection output created and then one or several collectionItems linked to the initial collection output

## HTTP Request

```
POST https://platform.taal.com/token-studio/api/v1/token/one-sat-ord/crea
```

## cURL

```
curl --location 'https://platform.taal.com/token-studio/api/v1/token/one-  
--header "ApiKey: ## VALID_API_KEY ##" \  
--header 'Content-Type: application/json' \  
--data '{  
    "projectId": "<projectId>",  
  
    "contentType": "text/html;charset=utf8",  
    "b64File": "PGh0bWw+PGJvZHk+SGVsbG8gV29ybGQ8L2JvZHk+PC9odG1sPg==",  
  
    "metadata": {  
        "name": "My single token 1718101145907",  
        "subType": "single",  
        "subTypeData": "{\"description\": \"this is description\" }",  
        "type": "ord",  
        "description": "this is a standalone token description",  
        "info": "anything",  
        "eyes": "green"  
    }  
}'
```

Response JSON structure

```
{  
    "success": true,  
    "data": {  
        "uid": <uid>,  
        "type": string,  
        "tokenProtocol": string  
    }  
}
```

## Body Parameters

Parameter	Description
subType	<ul style="list-style-type: none"> <li>'single' for standalone tokens in single projects</li> <li>'collection' for the seed token in collection projects           <ul style="list-style-type: none"> <li>'collectionItem' for each element linked to the collection</li> </ul> </li> </ul>
projectUid	projectUid
contentType	contentType depending on the desired output type (text, html or image)
b64File	Raw content formatted in b64
metadata	<p>Mandatory:</p> <pre>"name": NFT name "type": "ord" "subType": "single" //collection "subTypeData": stringified "description": string,</pre> <p>Other e.g. traits:</p> <pre>"info": "anything" "eyes": "green"</pre>

## Output deletion

Deletes an output (single, collection, collectinItem) in the project.

HTTP Request

```
DELETE https://platform.taal.com/token-studio/api/v1/output/{{outputUid}}
```

cURL

```
curl --location --request DELETE 'https://platform.taal.com/token-studio/
--header "Apikey: ## VALID_API_KEY ##" \
--header 'Content-Type: application/json'
```

## Response JSON structure

```
```json
"success": true,
  "data": {
    "uid": "<uid>",
    "createdAt": "<timestamp>",
    "type": "token",
    "tokenProtocol": "OneSatOrdinal",
    "projectUid": "<uid>",
    "transactionUid": null,
    "txId": null,
    "identityUid": "<uid>",
    "address": null,
    "metadata": { <metadata>
      },
    "offChainData": {
      <fileData>
    }
  }
}
```

```

## Query Parameters

| Parameter | Description       |
|-----------|-------------------|
| outputUid | A valid output Id |

# Output List

Returns all outputs for a valid project id.

## HTTP Request

```
GET https://platform.taal.com/token-studio/api/v1/output/by-project/{{pro
```

## cURL

```
curl --location 'https://platform.taal.com/token-studio/api/v1/' \
--header "Apikey: ## VALID_API_KEY ##"
```

## Response JSON structure

```
{  
  "success": boolean,  
  "data": {  
    "outputList": [  
      {  
        "createdAt": <date>,  
        "uid": <uid>,  
        "projectUid": <uid>,  
        "transactionUid": <uid> (null if has not been used in a transaction),  
        "type": string,  
        "tokenProtocol": string,  
        "metadata": { //Key / value pairs e.g.  
          "app": string,  
          "eyes": string,  
          "info": string,  
          "name": string,  
          "type": string,  
          "subType": string,  
          "description": string,  
          "subTypeData": stringified JSON  
        },  
        "offChainData": {  
          "b64FileHash": string,  
          "b64FileSize": int,  
          "contentType": string  
        }  
      }  
    ]  
  }  
}
```

## Query Parameters

| Parameter  | Description  |
|------------|--|
| projectUid | A valid project Id   |
| on-chain   | Optional. If provided, the accepted values are 0/1 (false / true) to filter by outputs already broadcast           |
| type       | Optional. If provided, the accepted values are collection / collectionItem, to filter by a specific type of output |

# Unsigned Transaction creation

Creates an unsigned transaction ready to sign and then be broadcast.

Unsigned transactions can refer to different outputs as well as different UTXO to be used for funding it

HTTP Request

```
POST https://platform.taal.com/token-studio/api/v1/token/one-sat-ord/crea
```

cURL

```
curl --location 'https://platform.taal.com/token-studio/api/v1/token/one-sat-ord/crea' \
--header "Apikey: ## VALID_API_KEY ##" \
--header 'Content-Type: application/json' \
--data '{
    "projectUid": "<projectUid>",
    "publicKey": <Public Key>,
    "dstAddress": <Token Recipient Address>,
    "outputList": [<outputList>],
    "utxoList": [
        {
            "outputIndex": num,
            "txId": "<txId>"
        }
    ]
}'
```

Response JSON structure

```
{
    "success": boolean,
    "data": {
        "transactionUid": <uid>,
        "txObj": {<txObj>}
    }
}
```

## Body Parameters

| Parameter    | Description                   |
|--------------|-------------------------------|
| <projectUid> | project Id                    |
| <publicKey>  | publicKey                     |
| dstAddress   | <Token Recipient Address>     |
| outputList   | List of output Ids            |
| txId         | Funding txId                  |
| outputIndex  | Output Index of the tx e.g. 1 |

## Broadcast Transaction

Broadcast a signed transaction and returns its transaction Id

### HTTP Request

```
POST https://platform.taal.com/token-studio/api/v1/submit
```

### cURL

```
curl --location 'https://platform.taal.com/token-studio/api/v1/submit' \
--header "Apikey: ## VALID_API_KEY ##" \
--header 'Content-Type: application/json' \
--data '{
    "transactionUid": "<transactionUid>",
    "tx": "<txObjSigned>"
}'
```

Response JSON structure

```
{  
  success: boolean,  
  data: {  
    txId: string  
  }  
}
```

## Body Parameters

| Parameter      | Description   |
|----------------|---|
| transactionUid | Transaction Uid returned by the create transaction endpoint |
| txObjSigned    | Signed transaction ready to be broadcast                    |



# Resources

# FAQ

## 1. How can I cancel a subscription?

Subscriptions last for a month and they are automatically renewed unless you specify it. If you don't want to renew one of your subscriptions:

- Go to "My Account"
- Unselect "Auto Renewal" for that subscription

## 2. What can I do when I can't find the subscription that I want to cancel?

- Go to "Support"
- Create a ticket for Taal Support team.

## 3. How can I view my purchase history?

- Go to "My Account"
- Click on "Invoicing & Billing"

## 4. How can I change my payment method?

- Go to "My Account"
- Click on "Invoicing & Billing" - "View More"
- Click on "Payment methods" - "Add payment method"
- Edit the desired subscription to use the new payment method.

## 5. How can I request a refund?

- Go to "Support"
- Create a ticket for Taal Support team.

## 6. How can I check the status of my refund request?

- Status of refund requests will be updated on the ticket

## 7. When will I get the money back?

- It takes about 10 business days for those valid refund cases. You will be kept informed on the ticket you raised for refund.

## 8. How can I get the invoice?

- Go to "My Account"
- Click on "Invoicing & Billing"

## 9. What happens if I don't pay TAAL Client subscription?

Subscription will be terminated and the corresponding API Key revoked.

## 10. Will TAAL charge me after free trial?

TAAL will only charge for the paid plans. To avoid limitations on Free keys, you will need to upgrade that specific key.

## 11. Can I upgrade my account?

In the circumstance where you change your subscription pursuant to the Transaction Processing Agreement, you must cancel your previous subscription directly with Stripe. Subscriptions through the TAAL services will continue to accrue automatic renewal charges via Stripe until such cancellation is received by Stripe.

## 12. When will I be billed for my subscription?

Billing happens at the time of subscription or renewal

## 13. When will my subscription be renewed?

Same day next month.

## 14. Can I pay with PayPal?

At the moment it is only possible to pay with Credit Card

### **15. What will happen if the remaining allowance for my plan is 0?**

The usage exceeding the allowance will be billed apart

# Support

## TAAL Community

Join our [TAAL Blockchain Discord](#) community for support on the Transaction Processing and STAS services.

Join our WhatsOnchain Telegram channel for support on the WhatsOnchain services: <https://t.me/joinchat/FfE6-EjZhoTHwhDhZH6F-w>

# Glossary

## Blockchain

It is a form of distributed database (or ledger) that acts as a record of all the valid transactions which are transmitted to the blockchain network.

## Contract

An on-chain document representing the contract between token issuer and user.

## Issue

Mint a token from an on-chain contract transaction.

## P2PKH (Pay to public key hash)

The most common locking script used by default to send bitcoin.

## Redeem

Destroy the token and return the locked Satoshis to the issuer.

## Satoshis

The smallest indivisible unit of a bitcoin. 1E8 Satoshis = 1 bitcoin.

## Script

Low level bitcoin script.

## SDK

Software Developer Kit is a software library to make certain tasks easier for a software developer.

## STAS

A token script named after its creator Stas. Also a reverse acronym: Substantiated Tokens from Actualized Satoshis.

## Token

It is a type of digital property or control element that can be transferred using a blockchain.

## UTXO

An Unspent Transaction Output is an amount of bitcoin that can be spent by the recipient.

# Acronyms and Abbreviations

| Acronym | Abbreviation                                  |
|---------|---|
| API     | Application Programming Interface             |
| BSV     | Bitcoin Satoshi Vision                        |
| JSON    | JavaScript Object Notation                    |
| MIME    | Multipurpose Internet Mail Extensions         |
| P2P     | Peer-to-peer transactions                     |
| P2PKH   | Pay to public key hash                        |
| SDK     | Software Developer Kit                        |
| SPV     | Simplified Payment Verification               |
| STAS    | Substantiated Tokens from Actualized Satoshis |
| TXID    | Transaction ID                                |
| UTXO    | Unspent Transaction Output                    |