WebSocket API

Connect to the Blocknative service over WebSocket protocol

Creating a Socket Connection

A WebSocket connection should be created with the Blocknative service atwss://api.blocknative.com/v0 . A successful connection will receive a response object that contains a status parameter with a value of'ok' .

Required Parameters

The following parameters are required for all messages that are sent to the Blocknative service:

parameter type description timeStamp string an ISO datetime string formattedYYYY-MM-DDThh:mm:ss.sTZD (see example for usage andhttps://www.w3.org/TR/NOTE-datetime for more details) dappId string your API key version string a version string usually used by the Blocknative SDK, but any version string can be used blockchain object object specifying a thesystem andnetwork of the blockchain (see next) system string blockchain name, valid value isethereum network string blockchain network, valid values for support systems are:

Ethereum (and EVM compatible) -main ,goerli ,kovan ,xdai ,matic-main ,

```

Copy { "timeStamp": "2021-01-11T06:21:40.197Z", "dappId": "", "version": "1", "blockchain": { "system": "ethereum" "network": "main" } // ...rest of the message specific parameter }

```

Initialization

Once a connection has been created with the Blocknative WebSocket service, an initialization message must be sent before any other messages so that the API key can be validated.

You API Key is created/managed from the Mempool Explorer account management page athttps://explorer.blocknative.com/account . Note that API Key is calleddappId in the Blocknative websocket protocol. ```

Copy { "categoryCode": "initialize", "eventCode": "checkDappId", // ...rest of the required parameters }

```

Subscriptions

You can subscribe to receive transaction state change events by either watching a transaction hash/id or an address.

Ethereum Transaction Hash

```

Copy // subscribe { "categoryCode":"activeTransaction", "eventCode":"txSent", "transaction": { "hash":"0x0b4c94c414f71ddd5e7a625fcaa83ff1f93e9a7ca37e0f577b488ac8fd786655" } // ...rest of the required parameters }

// unsubscribe { "categoryCode":"activeTransaction", "eventCode":"unwatch", "transaction": { "hash":"0x0b4c94c414f71ddd5e7a625fcaa83ff1f93e9a7ca37e0f577b488ac8fd786655" } // ...rest of the required parameters }

```

Address (Ethereum)

```

Copy // subscribe { "categoryCode":"accountAddress", "eventCode":"watch", "account": { "address":"0xf5b5bd85157587771db6d49315b1712201569bdd" } // ...rest of the required parameters }

// unsubscribe { "categoryCode":"accountAddress", "eventCode":"unwatch", "account": { "address":"0xf5b5bd85157587771db6d49315b1712201569bdd" } // ...rest of the required parameters }

```

Configurations
```

A configuration can be sent to the service to filter and/or decode transactions.

Transaction decoding is only available for Ethereum transactions There is a limit of 50 configurations per connection / API key. If you need to watch more than 50 addresses and also filter them, it is recommended that you subscribe to all of the addresses and then add a'global' scoped filter that will apply to all of those addresses. ```

Copy { "categoryCode":"configs", "eventCode":"put", "config": { "scope":"0xf5b5bd85157587771db6d49315b1712201569bdd", "filters":[{"contractCall.methodName":"transfer"}], "abi":[{ "constant":false, "inputs":[{"name":"newOwner","type":"address"}], "name":"setOwner", "outputs":[], "payable":false, "stateMutability":"nonpayable", "type":"function" }], "watchAddress":true } // ...rest of the required parameters }

```

parameter type description scope string A valid Ethereum address or'global' filters string[] An array of valid filters. The Blocknative service usesjsql , a JavaScript query language to filter events. Documentation for how to create filter queries can be foundhere abi string[] A valid JSON ABI that will be used to decode input data for transactions that occur on the contract address defined inscope watchAddress boolean defines whether the service should automatically watch the address as defined inscope scope establishes the context in which a filter is applied. Theglobal scope applies the filter to all transactions/addresses watched. Therefore aglobal scope only supports generic transaction elements. Ascope set to an address can include filters, such asmethodName, that are specific to the contract at that address. Similarly, theabi field only works for a contract specificscope .

Connection Drops

In the event of a WebSocket connection drop, no state is persisted between connections, so upon reconnection there will need to be logic defined that will:

1. Send an initialization message
2. Send all previously subscribed addresses, transactions and configs
3.

Transaction Events

Each message sent to the service will receive an echo of that message back, this echo may be ignored for most purposes. Every payload contains aneventCode parameter in theevent object which indicates the type of event. Detailed documentation for event codes can be found in thehere .

Ethereum Transaction Payload

```

Copy { "version":0, "serverVersion":"0.85.8", "timeStamp":"2021-01-06T01:25:24.716Z", "connectionId":"49babf8c03a18ca8577b8dc42e4ab9e8cbdf3148d4b4b1e311f54f24c19e0e57", "status":"ok", "event":{ "timeStamp":"2021-01-06T01:25:24.716Z", "categoryCode":"activeAddress", "eventCode":"txConfirmed", "dappId":"", "blockchain":{ "system":"ethereum", "network":"main" }, "transaction":{ "status":"confirmed", "monitorId":"GETH_1_F_PROD", "monitorVersion":"0.77.3", "timePending":"5482902", "blocksPending":410, "hash":"0x045fbad4090a25fc8267cf80f0f095a202cdac465d534e4c942b2068086e9aa3", "from":"0xbA99926Be550865a956febAF4A747FFA0648797A", "to":"0x111111125434b319222CdBf8C261674aDB56F3ae", "value":"0", "gas":324499, "gasPrice":"61000000000", "nonce":179, "blockHash":"0x543bd57151cb7ab3024a18a459844790278861c712e9e4b58e1939a207e9fb82", "blockNumber":11598068, "transactionIndex":307, "input": "0x34b0793b000000000000000...", "gasUsed": 138022, "asset":"ETH", "watchedAddress":"0x111111125434b319222cdbf8c261674adb56f3ae", "direction":"incoming", "counterparty":"0xbA99926Be550865a956febAF4A747FFA0648797A" } } }

```

Address subscription payload additional properties

If the transaction event is linked to an address subscription it will have the following properties that are unique to address subscriptions:

parameter type description watchedAddress string the address subscription that this transaction is linked to, allowing you to match the transaction to a subscription direction string a value ofincoming oroutgoing indicating whether the watched address is sending or receiving the transaction counterparty string the counter-party to the watched address

Confirmed/failed transaction payload additional properties

Transactions with a status ofconfirmed will have the following additional properties that are not included in other statuses:

parameter type description timePending string the time, in milliseconds, the transaction was pending before being included
```

in a block blocksPending string the number of blocks the transaction was pending before being included in a block blockHash string the hash of the block containing the confirmed (or failed) transaction blockNumber number the number of the block containing the confirmed (or failed) transaction gasUsed number the amount of gas used to confirm (or fail) the transaction

Replaced transactions payload additional properties

Transactions that have been replaced and have a status ofspeedup orcancel will have the following additional properties that are not included in other statuses:

parameter type description originalHash string the hash of the transaction that is being replaced

Errors

Errors from the service are returned as messages with astatus ='error' and a reason string explaining the error.

Websocket payload is too large (60,000 bytes):

```

Copy { //...other params status: 'error', reason: 'message too large' }

```

API key (dappId) parameter is missing:

```

Copy { //...other params status: 'error', reason: 'missing dappId' }

```

API key (dappId) is invalid:

```

Copy { //...other params status: 'error', reason: 'is not a valid API key' }

```

Attempt to watch an invalid address:

Address validity is based on the specific blockchainsystem used (bitcoin or ethereum). ```

Copy { //...other params "status":"error", "reason":"(stringaddress) 0 is an invalid address in objectaccount in object websocket-event" }

```

Amount of messages has exceeded limit, see Rate Limits section below.

```

Copy { //...other params status: 'error', reason: 'ratelimit', retryMs: 1000, blockedMsg:}

```

Exceeded daily event limit for your API key, see Rate Limits section below.

```

Copy { //...other params status: 'error', reason: 'You have reached your event rate limit for today. See account.blocknative.com for details.' }

```

IP has been rate limited, see Rate Limits section below.

```

Copy { //...other params status: 'error', reason: 'IP ratelimited' }

```
```

Error received for using incorrect API version.

```

Copy { //...other params status: 'error', message: 'api version not supported', }

```

Supported Networks

SeeMempool API Support Networks .

Rate Limits

Daily per API key event rate limits that apply to Webhook events also apply to WebSocket events. Details on those limits can be foundhere .

The Blocknative service also limits how many messages can be sent over a duration of time, limiting to 50 messages per second.

A strategy that the Blocknative SDK uses to handle these limits is to create a message queue that processes messages and ensures that messages are spaced adequately. This can be especially important to implement when re-sending addresses to be subscribed after a connection dropout.

If a rate limit error message is received you can use theretryMs parameter to wait for that time period to continue sending messages. If a message was blocked due to a rate limit it will be included in the payload in theblockedMsg parameter so you can re-send that message after the retry delay, ensuring that no messages are dropped.

Previous Supported Networks Next Ethernow Last updated6 months ago

Was this helpful?