

Join GitHub today

[Dismiss](#)

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)

Branch: master ▾

[grin](#) / [doc](#) / [api](#) / node_api.md[Find file](#)[Copy path](#)**jaspervdm** Optionally add rangeproofs to block api ([#2999](#))

0e62b7e 11 days ago

[4 contributors](#)

1154 lines (786 sloc) 30.3 KB

[Raw](#)[Blame](#)[History](#)

Node API Documentation

Table of Contents

1. [Blocks Endpoint](#)
 - i. [GET Blocks](#)
2. [Headers Endpoint](#)
 - i. [GET Headers](#)
3. [Chain Endpoint](#)

- i. [GET Chain](#)
 - ii. [POST Chain Compact](#)
 - iii. [GET Chain Validate](#)
 - iv. [GET Chain Outputs by IDs](#)
 - v. [GET Chain Outputs by Height](#)
- 4. [Status Endpoint](#)
 - i. [GET Status](#)
- 5. [TxHashSet Endpoint](#)
 - i. [GET TxHashSet Roots](#)
 - ii. [GET TxHashSet Last Outputs](#)
 - iii. [GET TxHashSet Last Range Proofs](#)
 - iv. [GET TxHashSet Last Kernels](#)
 - v. [GET TxHashSet Outputs](#)
 - vi. [GET TxHashSet Merkle Proof](#)
- 6. [Pool Endpoint](#)
 - i. [GET Pool](#)
 - ii. [POST Pool Push](#)
- 7. [Peers Endpoint](#)
 - i. [POST Peers Ban](#)
 - ii. [POST Peers Unban](#)
 - iii. [GET Peers All](#)
 - iv. [GET Peers Connected](#)
 - v. [GET Peers](#)

Blocks Endpoint

GET Blocks

Returns data about a specific block given a hash, a height or an unspent commit.

Optionally, Merkle proofs can be excluded from the results by adding `?no_merkle_proof`, rangeproofs can be included by adding `?include_proof` or results can be returned as "compact blocks" by adding `?compact`.

- **URL**

- `/v1/blocks/hash`
- `/v1/blocks/height`
- `/v1/blocks/commit`

- **Method:**

GET

- **URL Params**

Required: `hash=[string]` Or `height=[number]` Or `commit=[string]`

- **Data Params**

None

- **Success Response:**

- **Code:** 200
- **Content:**

Field	Type	Description
header	object	The block header

Field	Type	Description
- hash	string	Hash of the current block
- version	number	Version of the block
- height	number	Height of this block since the genesis block (height 0)
- previous	string	Hash of the block previous to this in the chain
- prev_root	string	Root hash of the header MMR at the previous header
- timestamp	string	RFC3339 timestamp at which the block was built
- output_root	string	Merklish root of all the commitments in the TxHashSet
- range_proof_root	string	Merklish root of all range proofs in the TxHashSet
- kernel_root	string	Merklish root of all transaction kernels in the TxHashSet
- nonce	number	Nonce increment used to mine this block
- edge_bits	number	Size of the cuckoo graph (2_log of number of edges)
- cuckoo_solution	[]number	The Cuckoo solution for this block
- total_difficulty	number	Total accumulated difficulty since genesis block
- secondary_scaling	number	Variable difficulty scaling factor for secondary proof of work
- total_kernel_offset	string	Total kernel offset since genesis block
inputs	[]string	Input transactions

Field	Type	Description
outputs	[]object	Outputs transactions
- output_type	string	The type of output Coinbase
- commit	string	The homomorphic commitment representing the output's amount (as hex string)
- spent	bool	Whether the output has been spent
- proof	string	Rangeproof (as hex string)
- proof_hash	string	Rangeproof hash (as hex string)
- block_height	number	Block height at which the output is found
- merkle_proof	string	Merkle proof
kernels	[]object	Transaction Kernels (a proof that a transaction sums to zero)
- features	object	Options for a kernel's structure or use
- bits	number	Representation of the features in bits
- fee	number	Fee originally included in the transaction this proof is for
- lock_height	number	The max lock_height of all inputs to this transaction
- excess	[]number	Remainder of the sum of all transaction commitments
- excess_sig	[]number	The signature proving the excess is a valid public key (signs the tx fee)

- **Error Response:**

- **Code:** 404 or 500
- **Content:** failed to parse input: Not found.

- **Sample Call:**

```
$.ajax({  
  url: "/v1/blocks/1",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

Headers Endpoint

GET Headers

Returns data about a block headers given either a hash or height or an output commit.

- **URL**

- /v1/headers/hash
- /v1/headers/height
- /v1/headers/commit

- **Method:**

GET

- **URL Params**

Required: hash=[string] OR height=[number] OR commit=[string]

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Field	Type	Description
header	object	The block header
- hash	string	Hash of the current block
- version	number	Version of the block
- height	number	Height of this block since the genesis block (height 0)
- previous	string	Hash of the block previous to this in the chain
- prev_root	string	Root hash of the header MMR at the previous header
- timestamp	string	RFC3339 timestamp at which the block was built
- output_root	string	Merklish root of all the commitments in the TxHashSet
- range_proof_root	string	Merklish root of all range proofs in the TxHashSet
- kernel_root	string	Merklish root of all transaction kernels in the TxHashSet
- nonce	number	Nonce increment used to mine this block

Field	Type	Description
- edge_bits	number	Size of the cuckoo graph (2_log of number of edges)
- cuckoo_solution	[]number	The Cuckoo solution for this block
- total_difficulty	number	Total accumulated difficulty since genesis block
- total_kernel_offset	string	Total kernel offset since genesis block

- **Error Response:**

- **Code:** 404 or 500
- **Content:** failed to parse input: Not found.

- **Sample Call:**

```
$.ajax({  
  url: "/v1/headers/1",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

Chain Endpoint

GET Chain

Retrieves details about the state of the current fork tip.

- **URL**

/v1/chain

- **Method:**

GET

- **URL Params**

None

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Field	Type	Description
height	number	Height of the tip (max height of the fork)
last_block_pushed	string	Last block pushed to the fork
prev_block_to_last	string	Block previous to last
total_difficulty	number	Total difficulty accumulated on that fork since genesis block

- **Error Response:**

- **Code:** 404 or 500

- **Content:** failed to parse input: Not found.

- **Sample Call:**

```
$.ajax({  
  url: "/v1/chain",  
  dataType: "json",  
  type : "GET",  
  success : function(r) {  
    console.log(r);  
  }  
});
```

POST Chain Compact

Trigger a compaction of the chain state to regain storage space.

- **URL**

/v1/chain/compact

- **Method:**

POST

- **URL Params**

None

- **Data Params**

None

- **Success Response:**

- **Code:** 200
- **Error Response:**
 - **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/chain/compact",  
  dataType: "json",  
  type : "POST",  
  success : function(r) {  
    console.log(r);  
  }  
});
```

GET Chain Validate

Trigger a validation of the chain state.

- **URL**

/v1/chain/validate

- **Method:**

GET

- **URL Params**

None

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Error Response:**

- **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/chain/validate",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

GET Chain Outputs By IDs

Retrieves details about specific outputs. Supports retrieval of multiple outputs in a single request.

- **URL**

- /v1/chain/outputs/byids?id=x
- /v1/chain/outputs/byids?id=x,y,z
- /v1/chain/outputs/byids?id=x&id=y&id=z

- **Method:**

GET

- **URL Params**

Required: id=[string]

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Field	Type	Description
outputs	[]object	Outputs
- output_type	string	The type of output Coinbase
- commit	string	The homomorphic commitment representing the output's amount (as hex string)
- spent	bool	Whether the output has been spent
- proof	string	Rangeproof (as hex string)
- proof_hash	string	Rangeproof hash (as hex string)
- block_height	number	Block height at which the output is found

Field	Type	Description
- merkle_proof	string	Merkle proof

- **Error Response:**

- **Code:** 404 or 500
- **Content:** []

- **Sample Call:**

```
$.ajax({  
  url: "/v1/chain/outputs/byids?id=0803516094a30830ed9fedff1c63251b51703ddffbb73f944d9e33e8fa5d17444f",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```



GET Chain Outputs By Height

Retrieves details about specific outputs. Supports retrieval of multiple outputs in a single request.

- **URL**

/v1/chain/outputs/byheight?start_height=x&end_height=y

- **Method:**

GET

- **URL Params**

Required: start_height=[number] end_height=[number]

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Field	Type	Description
header	object	The block header
- hash	string	Hash of the current block
- height	number	Height of this block since the genesis block (height 0)
- previous	string	Hash of the block previous to this in the chain
outputs	[]object	Outputs
- output_type	string	The type of output Coinbase
- commit	string	The homomorphic commitment representing the output's amount (as hex string)
- spent	bool	Whether the output has been spent
- proof	string	Rangeproof (as hex string)

Field	Type	Description
- proof_hash	string	Rangeproof hash (as hex string)
- block_height	number	Block height at which the output is found
- merkle_proof	string	Merkle proof

- **Error Response:**

- **Code:** 404 or 500
- **Content:** []

- **Sample Call:**

```
$.ajax({  
  url: "/v1/chain/outputs/byheight?start_height=101&end_height=200",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

Status Endpoint

GET Status

Returns various information about the node and the network

- **URL**

/v1/status

- **Method:**

GET

- **URL Params**

None

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Field	Type	Description
protocol_version	number	The node protocol version
user_agent	number	The node user agent
connections	number	The current number of connections
tip	object	The state of the current fork tip
height	number	Height of the tip (max height of the fork)
last_block_pushed	string	Last block pushed to the fork

Field	Type	Description
prev_block_to_last	string	Block previous to last
total_difficulty	number	Total difficulty accumulated on that fork since genesis block

- **Error Response:**

- **Code:** 404 or 500
- **Content:** failed to parse input: Not found.

- **Sample Call:**

```
$.ajax({  
  url: "/v1/stats",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

TxHashSet Endpoint

GET TxHashSet Roots

Retrieve the roots of the TxHashSet

- **URL**

/v1/txhashset/roots

- **Method:**

GET

- **URL Params**

None

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Field	Type	Description
output_root_hash	string	Output root hash
range_proof_root_hash	string	Rangeproof root hash
kernel_root_hash	string	Kernel set root hash

- **Error Response:**

- **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/txhashset/roots",
```

```
    dataType: "json",  
    type : "GET",  
    success : function(r) {  
        console.log(r);  
    }  
});
```

GET TxHashSet Last Outputs

Retrieves the last n outputs inserted into the tree.

- **URL**
 - /v1/txhashset/lastoutputs (gets last 10)
 - /v1/txhashset/lastoutputs?n=x

- **Method:**

GET

- **URL Params**

Required: n=[number]

- **Data Params**

None

- **Success Response:**

- **Code:** 200
- **Content:**

Array of

Field	Type	Description
hash	string	hash of the outputs

- **Error Response:**

- **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/txhashset/lastoutputs?n=20",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

GET TxHashSet Last Range Proofs

Retrieves the last n rangeproofs inserted in to the tree.

- **URL**

- /v1/txhashset/lastrangeproofs (gets last 10)
 - /v1/txhashset/lastrangeproofs?n=x

- **Method:**

GET

- **URL Params**

Required: n=[number]

- **Data Params**

None

- **Success Response:**

- **Code:** 200
- **Content:**

Array of

Field	Type	Description
hash	string	hash of the rangeproofs

- **Error Response:**

- **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "v1/txhashset/lastrangeproofs",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

GET TxHashSet Last Kernels

Retrieves the last n kernels inserted in to the tree.

- **URL**

- /v1/txhashset/lastkernels (gets last 10)
- /v1/txhashset/lastkernels?n=x

- **Method:**

GET

- **URL Params**

Required: n=[number]

- **Data Params**

None

- **Success Response:**

- **Code:** 200
- **Content:**

Array of

Field	Type	Description
hash	string	hash of the kernels

- **Error Response:**

- **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/txhashset/lastkernels?n=20",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

GET TxHashSet Outputs

UTXO traversal. Retrieves last utxos since a start index until a max.

- **URL**

/v1/txhashset/outputs?start_index=x&max=y

- **Method:**

GET

- **URL Params**

Required: start_index=[number] max=[number]

- **Data Params**

None

- **Success Response:**

- **Code:** 200

◦ **Content:**

Field	Type	Description
highest_index	number	The last available output index
last_retrieved_index	number	The last insertion index retrieved
outputs	[]object	Outputs
- output_type	string	The type of output Coinbase
- commit	string	The homomorphic commitment representing the output's amount (as hex string)
- spent	bool	Whether the output has been spent
- proof	string	Rangeproof (as hex string)
- proof_hash	string	Rangeproof hash (as hex string)
- block_height	number	Block height at which the output is found
- merkle_proof	string	Merkle proof
kernels	[]object	Transaction Kernels (a proof that a transaction sums to zero)
- features	object	Options for a kernel's structure or use
- bits	number	Representation of the features in bits
- fee	number	Fee originally included in the transaction this proof is for
- lock_height	number	The max lock_height of all inputs to this transaction
- excess	[]number	Remainder of the sum of all transaction commitments

Field	Type	Description
- excess_sig	[]number	The signature proving the excess is a valid public key (signs the tx fee)

- **Error Response:**

- **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/txhashset/outputs?start_index=1&max=100",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

GET TxHashSet Merkle Proof

Build a merkle proof for a given output id and return a dummy output with merkle proof for position filled out.

- **URL**

/v1/txhashset/merkleproof?id=x

- **Method:**

GET

- **URL Params**

Required: id=[string]

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Field	Type	Description
outputs	[]object	Outputs
- output_type	string	The type of output Coinbase
- commit	string	The homomorphic commitment representing the output's amount (as hex string)
- spent	bool	Whether the output has been spent
- proof	string	Rangeproof (as hex string)
- proof_hash	string	Rangeproof hash (as hex string)
- block_height	number	Block height at which the output is found
- merkle_proof	string	Merkle proof

- **Error Response:**

- **Code:** 404 or 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/txhashset/merkleproof?id=0803516094a30830ed9fedff1c63251b51703ddffbb73f944d9e33e8fa5d1744",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```



Pool Endpoint

GET Pool

Retrieves basic information about the transaction pool.

- **URL**

/v1/pool

- **Method:**

GET

- **URL Params**

None

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Field	Type	Description
pool_size	number	Number of transactions in the memory pool

- **Error Response:**

- **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/pool",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

POST Pool Push

Push new transaction to our local transaction pool. Add `?fluff` at the end of the URL to bypass Dandelion relay.

- **URL**

/v1/pool/push

- **Method:**

POST

- **URL Params**

None

- **Data Params**

file=[string] (hex encoded transaction)

- **Success Response:**

- **Code:** 200

- **Error Response:**

- **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/pool/push",  
  dataType: "json",  
  type: "POST",  
  data: {  
    file: tx  
  },  
  success: function(r) {  
    console.log(r);  
  }  
});
```

Peers Endpoint

POST Peers Ban

Ban a specific peer.

- **URL**

`/v1/peers/a.b.c.d:p/ban`

- **Method:**

`POST`

- **URL Params**

`a.b.c.d:p=[string]`

- **Data Params**

`None`

- **Success Response:**

- **Code:** 200

- **Error Response:**

- **Code:** 400

- **Sample Call:**

```
$.ajax({  
  url: "/v1/peers/192.168.1.1:13414/ban",  
  dataType: "json",  
  type: "POST",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

POST Peers Unban

Unban a specific peer.

- URL

/v1/a.b.c.d:p/unban

- Method:

POST

- URL Params

a.b.c.d:p=[string]

- Data Params

None

- Success Response:

- Code: 200

- Error Response:

- **Code:** 400

- **Sample Call:**

```
$.ajax({  
  url: "/v1/192.168.1.1:13414/unban",  
  dataType: "json",  
  type: "POST",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

GET Peers All

Retrieves all peers in db.

- **URL**

/v1/peers/all

- **Method:**

GET

- **URL Params**

None

- **Data Params**

None

- **Success Response:**

- **Code:** 200
- **Content:**

Array of

Field	Type	Description
addr	string	Network address of the peer
capabilities	object	What capabilities the peer advertises
- bits	number	Representation of the capabilities in bits
user_agent	string	The peer user agent
flags	string	State the peer has been detected with.
last_banned	number	The time the peer was last banned
ban_reason	string	The reason for the ban

- **Error Response:**

- **Code:** 500

- **Sample Call:**

```
$.ajax({  
  url: "/v1/peers/all",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```

GET Peers Connected

Retrieves all connected peers

- **URL**

/v1/peers/connected

- **Method:**

GET

- **URL Params**

None

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Array of

Field	Type	Description
capabilities	object	What capabilities the peer advertises
- bits	number	Representation of the capabilities in bits

Field	Type	Description
user_agent	string	The peer user agent
version	number	Software version of the peer
addr	string	Network address of the peer
total_difficulty	number	Total of difficulty of the peer
height	number	Height of the peer
direction	string	Direction of the connection (Inbound

- **Error Response:**

- **Code: 500**

- **Sample Call:**

```
$.ajax({  
  url: "/v1/peers/connected",  
  dataType: "json",  
  type : "GET",  
  success : function(r) {  
    console.log(r);  
  }  
});
```

GET Peers

Retrieves information about a specific peer.

- **URL**

/v1/peers/a.b.c.d

- **Method:**

GET

- **URL Params**

a.b.c.d=[string]

- **Data Params**

None

- **Success Response:**

- **Code:** 200

- **Content:**

Field	Type	Description
addr	string	Network address of the peer
capabilities	object	What capabilities the peer advertises
- bits	number	Representation of the capabilities in bits
user_agent	string	The peer user agent
flags	string	State the peer has been detected with.
last_banned	number	The time the peer was last banned
ban_reason	string	The reason for the ban

- **Error Response:**
 - **Code:** 404 or 500
- **Sample Call:**

```
$.ajax({  
  url: "/v1/peers/192.168.1.2",  
  dataType: "json",  
  type: "GET",  
  success: function(r) {  
    console.log(r);  
  }  
});
```