**Builder API**

The following describes the API endpoints a builder needs to support in order to build blocks in Espresso.

These API endpoints are actively maintained and might change going forward.

**GET block_info/availableblocks/:parent_hash/:view_number/:sender/:signature:**

This endpoint allows a sequencer node to query a builder for a list of available blocks that the builder can provide, ensuring consistency with the ongoing consensus defined by the specified parent_hash.

**Parameters**

Name

Type

Description

parent_hash

TaggedBase64

The hash of the parent block in the current consensus chain. This parameter ensures that the returned blocks are valid continuations of the existing chain.

view_number

Integer

View number in Hotshot.

sender

TaggedBase64

The address or identifier of the entity making the request (i.e., the sequencer node).

signature

TaggedBase64

A cryptographic signature generated by the sequencer node, authenticating the request and proving its authenticity

**Returns**

Copy

```
[
  {
    "block_hash": TaggedBase64,
    "block_size": integer,
    "offered_fee": integer,
```

```
    "signature": TaggedBase64,
    "sender": TaggedBase64
  }
]
```

**GET block_info/claimblock/:block_hash/:view_number/:sender/:signature:**

Upon receiving the metadata about available blocks through the previously mentioned endpoint, the sequencer node will utilize this dedicated endpoint to retrieve the complete and comprehensive content of *chosen* block. This endpoint will provide the sequencer node with the entirety of the block data, including all transactions, consensus data, and any other pertinent information encapsulated within the block. By leveraging this endpoint, the sequencer node can access and process the complete block content, enabling it to perform its designated functions effectively inside the Hotshot consensus.

Parameters

Name

Type

Description

block_hash

TaggedBase64

The hash of the block provided as a response in previous API.

view_number

Integer

View number in Hotshot.

sender

TaggedBase64

The address or identifier of the entity making the request (i.e., the sequencer node).

signature

TaggedBase64

A cryptographic signature generated by the sequencer node, authenticating the request and proving its authenticity

**Returns**

Copy

```
{
  "block_payload": Payload,
```

```
  "metadata": MetaData,
  "signature": TaggedBase64,
  "sender": TaggedBase64
}
```

**GET block_info/claimheaderinput/:block_hash/:view_number/:sender/:signature**

This endpoint enables a sequencer node to retrieve the relevant fee information associated with a specific block, facilitating a transparent payout mechanism for the builder responsible for constructing that block.

**Note:** The block_hash should match the one requested in previous API i.e claim_block for consistent fee payoff.

Parameters

| Name | Type | Description |
| --- | --- | --- |
| block_hash | TaggedBase64 | The hash of the block provided as a response in first API. |
| view_number | Integer | View number in Hotshot. |
| sender | TaggedBase64 | The address or identifier of the entity making the request (i.e., the sequencer node). |
| signature | TaggedBase64 | A cryptographic signature generated by the sequencer node, authenticating the request and proving its authenticity |

**Returns**

Copy

```
{
  "vid_commitment": VidCommitment,
  "vid_precompute_data": VidPrecomputeData,
```

"fee_signature": TaggedBase64,

"signature": "TaggedBase64,

"sender" = TaggedBase64

}

**POST /txn_submit/submit**

It enables external user to submit directly to builder's private transactions mempool.

- Returns the hash of the transaction if it was successfully submitted. This does not mean the transaction has yet been sequenced. The user can check for inclusion of the transaction using /availability/transaction/hash/:hash from the availability API.
    - Request Body Transaction
    - Returns tagged<TX>

**POST /txn_submit/batch**

It enables external user to submit a list of transactions to builder's private transactions mempool.

- Returns the corresponding list of transaction hashes if it were successfully submitted. This does not mean the transaction has yet been sequenced. The user can check for inclusion of the transaction using /availability/transaction/hash/:hash from the availability API.
    - Request Body Vec<Transaction>
    - Returns Vec<tagged<TX>>

**GET /txn_submit/status/:transaction_hash**

It enables external user to track the status of submitted transactions in short period, old transaction will be pruned if the map doesn't have enough space.

- Returns the status of the queried transaction commitment if successfully retrieved. The status will be one of the following: "Pending", "Unknown" or "Rejected" (with a reason message). The status "Sequenced" is not yet supported. While this information can be obtained from the query service or block explorer, transactions with a "Sequenced" status will appear as "Pending" when retrieved. An error will be returned if the request fails due to issues like malformed input, incorrect deserialization, or missing data.
    - Request Body tagged<TX>

**Returns**

Copy

{ "Pending" }

or

Copy

{ "Unknown" }

or

Copy

{ "Rejected": { "reason": TaggedBase64 } }

or

Copy

{ "error": TaggedBase64 }