Avail Light Client reference

How to Configure a Light Client

The Avail light client isheavily customisable with the most important parameters being exposed through its config file, the full config reference along with its default values can be foundhere(opens in a new tab) . Light client config example:

log_level = "info" http_server_host = "127.0.0.1" http_server_port = "7000"

port = "37000" ot_collector_endpoint = "http://0.0.0.0:4317"

full_node_ws = ["ws://127.0.0.1:9944"] app_id = 0 confidence = 92.0 avail_path = "avail_path"

Set to actual bootstrap peer ID and multiaddress

bootstraps = [["12D3KooWStAKPADXqJ7cngPYXd2mSANpdgh1xQ34aouufHA2xShz", "/ip4/127.0.0.1/tcp/39000"]]

Application client

The app client mode is used by individual apps to download, reconstruct and locally store relevant app data. Application data is primarily downloaded from the DHT, either by downloading individual block matrix rows or with per-cell approach, downloading relevant individual cells. RPC is (again) used as a fallback mechanisms, if DHT doesn't contain the data.

Downloaded and reconstructed data is exposed through a HTTP endpoint, with port configured by thehttp_server_port parameter.

App client mode is activated by setting theapp_id to a value greater than0 .

How to Embed Your Light Client

Fetching the number of the latest block processed by light client

To fetch the number of the latest block processed by light client, we can performGET request on/v1/latest_block endpoint.

CURL Rust curl

"http://localhost:7000/v1/latest_block" Response:

{ "latest_block" :

10 }

Fetching the confidence for given block

To fetch the confidence for specific block, which is already processed by application client, we can performGET request on/v1/confidence/{block_number} endpoint.

CURL Rust curl

"http://localhost:7000/v1/confidence/1" Response:

{ "block" :

1 , "confidence" :

93.75 , "serialised_confidence" :

"5232467296" }

Fetching decoded application data for given block

After data is verified, it can be fetched withGET request on/v1/appdata/{block_number} endpoint, by specifyingdecode=true query parameter.

CURL Rust

JSON response

curl

"http://localhost:7000/v1/appdata/1?decode=true" Response:

{ "block" :

46 , "extrinsics" : ["ZXhhbXBsZQ=="] }

Decoded extrinsic

curl

-s

"http://127.0.0.1:7000/v1/appdata/1?decode=true"

|

jq

-r

'.extrinsics[-1]'

|

base64

-d Response:

"example"

Waiting for application data to be verified

If light client is still processing specific block, we can poll light client withGET request on/v1/appdata/{block_number} endpoint, and wait for data to become available.

CURL Rust curl

"http://localhost:7000/v1/appdata/2" If response status code is401 and body is:

"Processing block" we need to poll endpoint until data becomes available.

API reference

In case of error, endpoints will return response with500 Internal Server Error status code, and descriptive error message.

GET

/v1/mode

Retrieves the operating mode of the light client. Light client can operate in two different modes,LightClient orAppClient , depending on configuration of application ID.

Responses

If operating mode isLightClient response is:

Status code:200 OK "LightClient" In case ofAppClient mode, response is:

Status code:200 OK { "AppClient" : {app_id}}

GET

/v1/latest_block

Retrieves the latest block processed by the light client.

Responses

Status code:200 OK { "latest_block" : {block_number}}

GET

/v1/confidence/{block_number}

Given a block number, it returns the confidence computed by the light client for that specific block.

Path parameters: * block_number * - block number (required)

Responses

In case when confidence is computed:

Status code:200 OK { "block" :

1 ,

"confidence" :

93.75 ,

"serialised_confidence" :

"5232467296" } If confidence is not computed, and specified block is before the latest processed block:

Status code:400 Bad Request "Not synced" If confidence is not computed, and specified block is after the latest processed block:

Status code:404 Not Found "Not found"

GET

/v1/appdata/{block_number}

Given a block number, it retrieves the hex-encoded extrinsics for the specified block, if available. Alternatively, if specified by a query parameter, the retrieved extrinsic is decoded and returned as a base64-encoded string.

Path parameters: * block_number * - block number (required)

Query parameters: * decode * -true * if decoded extrinsics are requested (boolean, optional, default isfalse *)

Responses

If application data is available, and decode isfalse or unspecified:

Status code:200 OK { "block" :

1 , "extrinsics" : [
"0xc5018400d43593c715fdd31c61141abd04a99fd6822c8558854ccde39a5684e7a56da27d01308e88ca257b65514b7b44fc1913a6a9af6abc34c3d22761b0e425674d68df7de26be1c8533a7bbd01fdb3a8
] } If application data is available, and decode istrue :

Status code:200 OK { "block" :

1 ,

"extrinsics" : ["ZXhhbXBsZQ=="] } If application data is not available, and specified block is the latest block:

Status code:401 Unauthorized "Processing block" If application data is not available, and specified block is not the latest block:

Status code:404 Not Found "Not found"

GET

/v1/status/{block_number}

Retrieves the status of the latest block processed by the light client.

Path parameters: * block_number * - block number (required)

Responses

If latest processed block exists, andapp_id is configured (otherwise,app_id is not set):

Status code:200 OK { "block_num" :

89 ,

"confidence" :

93.75 ,

"app_id" :

1 } If there are no processed blocks:

Status code:404 Not Found "Not found"

Run a Light Client Run an Avail node