Guide to the æternity middleware (mdw) and API generally

author: John Newby
date: 2019-07-01
corresponds to version: v0.6.1

Introduction

The middleware, mdw, is a server process which sits in front of the æternity node ('node'), performing various functions

- caching often-used results from the node, and returning them more quickly than the node can
- returning results, usually aggregated, which the node cannot.

When starting up, mdw queries the node in order to acquire a representation of the blockchain, which it stores in a SQL database. While storing these data mdw also makes other queries and stores extra data, for example

- for contract calls, mdw decodes the arguments and return value, when possible
- for oracle creation, mdw calculates the oracle id and stores that, in order that it can list oracles, and associate them with their requests and responses.
- mdw keeps a list of registered names and their expiration heights, and will do a reverse lookup to complement the forward lookup which the node supports.

In general when we find a useful query which is not supported by the node, we attempt to implement it. And you can use SQL queries to make up the rest.

Installation, or not.

mdw is written in the Rust language, and is relatively straightforward to install. We, the æternity team host instances for both mainnet and testnet, which you're welcome to use. We do understand though that our users may not want to trust us, in which case you'll be wanting to run your own instance. You can start off by priming with our database--we expose a read-only replica of the PostgreSQL database which we also expose. You may verify this backup using the method described in the 'verification' section, below.

The installation instructions are in the README.md file on the Github repository for the project, at https://github.com/aeternity/aepp-middleware

Interfaces

mdw has an HTTP API, and a Websocket API. The Websocket API permits subscriptions to events of interest, for instance key block, micro block or transaction generation, or a subscription to any type of object in the block chain. In particular the Websocket API is intended for use cases such as

- looking for queries to an Oracle, possibly in order to generate responses
- being notified of calls to a contract
- finding payments to, or from an account of interest

The HTTP REST API complements the node API and allows more expensive queries than the node itself supports.

Best practises running mdw

Broadly, mdw has two modes of operation--population, when it loads the blockchain into the database, and serving, which exposes the database via a REST API and forwards requests to the node behind it via HTTP. You may run an instance of mdw in one, or both of these

modes. For a simple installation, the simplest way to do this is to run both, with the '-s -p' options. If you're running a service which needs to be constantly available, it will be more sensible to run one population instance, and one or more serving instances. These could run from a read-only replica of the database, and many can run in parallel if scaling is required.

In the scripts/ directory you will find systemd scripts which can be modified for these use-cases, and a python script which monitors the database and restarts mdw if the top block is more than 10 minutes old. We're working hard to squash all bugs, but we're not there yet. However using the monitoring script in conjunction with systemd gives very high reliability for very little effort.

mdw has very light hardware requirements--it can run perfectly well, along with its database from a Raspberry Pi. As you would expect, the more hardware you throw at it, the happier it is, though.

How to use mdw

The middleware is a layer over the node, and as such presents the node behind it more or less as-is, with the exception of some node endpoints which it proxies. mdw attempts to give a coherent view of the node--for instance when it is still loading blocks from the database it will reports its current generation as the highest in the database, rather than the highest the node reports. And whilst catching up, mdw will report via the websocket interface all objects loaded in. In this way, one can use the notifications semi-asynchronously.

HTTP interface

State channels

`GET /middleware/channels/active`
returns a list of all state channels which have been opened, and not closed. Example:

```
$ curl https://mdw.aepps.com/middleware/channels/active
["ch_2tceSwiqxgBcPirX3VYgW3sXgQdJeHjrNWHhLWyfZL7pT4gZF4","ch_AG5wzf4F9nMyuAmPav981D
k2XiQhAFvWAiNbUniZNPvk1qZxa","ch_DhA1FvZ2vcN9waEUmTcztPbjYH8aJ6FcKiLEUg5xWYbm9ktSU"
,"ch_2KP1gKWTgFxmPWpQDWr1Gbghi18ZtxPMFELqhEQ651B2a5ZtXi","ch_2jAjhyQ4kTpuJbANBDMWtd
saDyFjAaAEmoVFmuFvKXnZLvt7hn","ch_284sMWGcDzkf6LGbZVkNwmt2rieeLdPtSJCtVd7QQpTuoZHMk
Q","ch_29yF8QQuKgk2ngJ36bkfqD5aXuNC5PrSHvgaUKFYWAvFqXF4db","ch_2gs8b8LUa5HgmPSEmCpE
NM5UPYQXGpAZx8rDSTvZdtv1C9QseC"]
```

`GET /middleware/channels/transactions/address/<address>`
for this state channel, show its on-chain transactions:

```
$ curl -s
https://mdw.aepps.com/middleware/channels/transactions/address/ch_2tceSwiqxgBcPirX3
VYgW3sXgQdJeHjrNWHhLWyfZL7pT4gZF4|jq
{
  "transactions": [
    {
```

```
      "block_height": 9155,
      "block_hash": "mh_2C1TrRnqvc8tAyeRpj6YWeuZHCgST1p5uackmJJ5VdyP3rrGMT",
      "hash": "th_2kXfesqmRusaiN8CzhjizXztC41TbdMb8EqvKMKJWTnCcfwvrY",
      "signatures": [

"sg_ADWpdrNBXGX9f245Pu8RLDQ5AeRQCiw8UyUrQKoFMkvgUSULNuZPAAo1tvfhyusRgHKtW5Q92hzoqj1
MZcVH4ub7dswJy",

"sg_LRsP3UzUnUb4TtXqPDmUmQsb2aB1GUV8g3JMF4mWpfcChtBuyGtbzFpJfTYkKpvMEpc6AMJM156bCr2
4BNtUk75NNGJLK"
      ],
      "tx": {
        "channel_reserve": 2,
        "delegate_ids": [],
        "fee": 20000,
        "initiator_amount": 10,
        "initiator_id": "ak_2VsncWAk9qkA8SAY8zpcympSaCN313TV9GjAPZ9XQUFMSz4vTf",
        "lock_period": 10,
        "nonce": 2,
        "responder_amount": 10,
        "responder_id": "ak_25UPgAhVxTrq6CCyjDYhMpPadW6QLHNxtV5a2je12RGk1Rfmjt",
        "state_hash": "st_AkEG+wvKWZdW9R+Zzz+7HHTTR3KWcTNrQNpGMr/VmR3DqtiC",
        "type": "ChannelCreateTx",
        "version": 1
      }
    }
  ]
}
```


GET /middleware/contracts/all

all contracts, most recent first. Optionally page and limit:


```
$ curl -s 'https://mdw.aepps.com/middleware/contracts/all?limit=2&page=1'|jq
[
  {
    "block_height": 97934,
    "contract_id": "ct_AhMbfHYPBK8Qu1DkqXwQHcMKZoZAUndyYTNZDnyS1AdWh7X9U",
    "transaction_hash": "th_2raHdPQ8xtbE6oKh3z1pFmUpyFC5H7ZTBkNB8TuVydJjwedduL"
  },
  {
    "block_height": 94121,
    "contract_id": "ct_DQwmk4nuAJz2aBxBkhz4xMck8M7SnK8iUXGi4cGGhyU5BMRjA",
    "transaction_hash": "th_pAR94ZVXKrBfXcG5Z6cAUGd76DSAn2ESGdtkeVN89jXkXJUEP"
  }
]
```


GET /middleware/contracts/calls/address/<address>

If the contract has calls, this endpoint returns them, if mdw has access to the HTTP compiler it will also have unpacked the arguments and return value and returns them.


```
$ curl -s
'https://mdw.aepps.com/middleware/contracts/calls/address/ct_AhMbfHYPBK8Qu1DkqXwQHc
MKZoZAUndyYTNZDnyS1AdWh7X9U' | jq
[
```

```
  {
    "arguments": {
      "arguments": [
        {
          "type": "list",
          "value": [
            {
              "type": "tuple",
              "value": [
                {
                  "type": "word",
                  "value": 1.498635465619252e+76
                },
                {
                  "type": "word",
                  "value": 60000000000
                }
              ]
            }
          ]
        }
      ],
      "function": "payout"
    },
    "caller_id": "ak_2vx4yNy2FUi7Fe2ZjKbKvpnabDTJE8RijtfAhQHNjY5zfj1We6",
    "callinfo": {
      "caller_id": "ak_2vx4yNy2FUi7Fe2ZjKbKvpnabDTJE8RijtfAhQHNjY5zfj1We6",
      "caller_nonce": 2,
      "contract_id": "ct_AhMbfHYPBK8Qu1DkqXwQHcMKZoZAUndyYTNZDnyS1AdWh7X9U",
      "gas_price": 1000000000,
      "gas_used": 22150,
      "height": 97941,
      "log": [],
      "return_type": "ok",
      "return_value": "cb_AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADfhHWADzMnYB"
    },
    "contract_id": "ct_AhMbfHYPBK8Qu1DkqXwQHcMKZoZAUndyYTNZDnyS1AdWh7X9U",
    "transaction_id": "th_NVCNN7txvnDJwT9S8Qb13ffX3E6GcfmfBLhXco8AwLLDZgpHm"
  }
]
```


`GET /middleware/contracts/transactions/address/<address>`

All transactions for this contract

```
$ curl -s
'https://mdw.aepps.com/middleware/contracts/transactions/address/ct_AhMbfHYPBK8Qu1D
kqXwQHcMKZoZAUndyYTNZDnyS1AdWh7X9U' | jq
```


```
{
  "transactions": [
    {
      "block_height": 97934,
      "block_hash": "mh_bbAzJcovSNLNW1qwPMWkofdRsXShnQcHFzscyxLp7gxhoHuZB",
      "hash": "th_2raHdPQ8xtbE6oKh3z1pFmUpyFC5H7ZTBkNB8TuVydJjwedduL",
      "signatures": [
```

"sg_Q3Ud2LKftTKKKCCWfEFLUs1THsczRS9xJjFvxowDBcvay3azYfyRumaF2aQnqSpHkWZ1GBBE2wbMUA7
NBgnEinmLKFwyN"
      ],
      "tx": {
        "abi_version": "0x0001",
        "amount": 0,
        "call_data":
"cb_AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACC5yVbyizFJqfWYeqUF89obIgnMVzkjQAYrtsG
9n5+Z6gAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAnHQYrA==",
        "code":
"cb_+QhpRgGgqQNVd4r2/yyoTbBfj5MFzMC1UeBDQro+Ve+ke+6Ev7P5Bs/5Ak2gqmFCUiigORZ09u2r5+4
7cRipSBbbt8qP5wUU5XnLwWiGcGF5b3V0uQHgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABgA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAKD///
/wAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAOAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAEgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAWAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAABgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAcD//wAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAuEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAIAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA+QHLoLnJVvKLMUmp9Zh6pQXz2hsiCcxXOSNABiu2wb2fn5nqhGluaXS4YAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAP/////
/////////////////////////////////////7kBQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
gAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAYAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAMAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAEA//8AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAA///+QKuoPo8c94emJ3PC7jHvg6QmcI3Ep
tLCBFzx9KYmzO4otqLh3BheW91dCe5AkAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAIAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAoAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHgAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA4A
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
SAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABYAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAABwP//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACIP///////////////////////////////
///////////AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAC4QAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAC5AXFiAACPYgAAr5GAgIBRf7n
JVvKLMUmp9Zh6pQXz2hsiCcxXOSNABiu2wb2fn5nqFGIAAUtXUICAUX+qYUJSKKA5FnT27avn7jtxGKlIFt
u3yo/nBRTlecvBaBRiAADaV1CAUX/6PHPeHpidzwu4x74OkJnCNxKbSwgRc8fSmJszuKLaixRiAAFXV1BgA
RlRAFtgABlZYCABkIFSYCCQA2ADgVKQWWAAUVlSYABSYADzW2AAgFJgAPNbWVlgIAGQgVJgIJADYAAZWWAg
AZCBUmAgkANgA4FSgVKQVltgIAFRUZBQWVCAkVBQgGAAkJFQW4GAYAEBYgABAFdQgJFQUJBWW4BgAQFiAAE
QV1BgARlRAFuAUYBRkGAgAVGRYCABUWAAYABgAIRZYCABkIFSYCCQA2ABgVKGYABa8VCAg4UBlFCUUFBQUG
IAAO5WW1BQgpFQUGIAALdWW2AgAVGAUZBgIAFRWVCBgZJQklBQYgAA7lYqt7+f",
        "deposit": 0,
        "fee": 1400000000000000,
        "gas": 100000,
        "gas_price": 1000000000,
        "nonce": 1,
        "owner_id": "ak_2vx4yNy2FUi7Fe2ZjKbKvpnabDTJE8RijtfAhQHNjY5zfj1We6",
        "type": "ContractCreateTx",
        "version": 1,

```
        "vm_version": "0x0004"
      }
    },
    {
      "block_height": 97941,
      "block_hash": "mh_2JrTVhpsHUzuJ8CcFZduXzFUCg3D6xpxNQTop7xYHVoUrHvsHr",
      "hash": "th_NVCNN7txvnDJwT9S8Qb13ffX3E6GcfmfBLhXco8AwLLDZgpHm",
      "signatures": [

"sg_HTFBtCKZoQDmwAYnnJSfe4x2yAeGc67ua6seZ3dKpsFEWLqBhBE9AvTk7wWNdMsa9DDt2FkkvE9qJqq
RD6RKjYiBAwqm7"
      ],
      "tx": {
        "abi_version": "0x0001",
        "amount": 60000000000,
        "call_data":
"cb_AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACCqYUJSKKA5FnT27avn7jtxGKlIFtu3yo/nBRT
lecvBaAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAwP/////////////////////////
////////////ISH55Kvb3fVQMHE6sZ4ggE/hws0iDLouYXBoCtN3D+4AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAN+EdYAKZBmxk=",
        "caller_id": "ak_2vx4yNy2FUi7Fe2ZjKbKvpnabDTJE8RijtfAhQHNjY5zfj1We6",
        "contract_id": "ct_AhMbfHYPBK8Qu1DkqXwQHcMKZoZAUndyYTNZDnyS1AdWh7X9U",
        "fee": 458800000000000,
        "gas": 458800,
        "gas_price": 1000000000,
        "nonce": 2,
        "type": "ContractCallTx",
        "version": 1
      }
    }
  ]
}
```


GET /middleware/generations/<from>/<to>?<limit>&<page>

All


```
$ curl -s 'https://mdw.aepps.com/middleware/generations/1/2' | jq
{
  "data": {
    "1": {
      "beneficiary": "ak_2RGTeERHPm9zCo9EsaVAh8tDcsetFSVsD9VVi5Dk1n94wF3EKm",
      "hash": "kh_29Gmo8RMdCD5aJ1UUrKd6Kx2c3tvHQu82HKsnVhbprmQnFy5bn",
      "height": 1,
      "micro_blocks": {
        "mh_ufiYLdN8am8fBxMnb6xq2K4MQKo4eFSCF5bgixq4EzKMtDUXP": {
          "hash": "mh_ufiYLdN8am8fBxMnb6xq2K4MQKo4eFSCF5bgixq4EzKMtDUXP",
          "pof_hash": "no_fraud",
          "prev_hash": "kh_29Gmo8RMdCD5aJ1UUrKd6Kx2c3tvHQu82HKsnVhbprmQnFy5bn",
          "prev_key_hash": "kh_29Gmo8RMdCD5aJ1UUrKd6Kx2c3tvHQu82HKsnVhbprmQnFy5bn",
          "signature":
"sg_91zukFywhEMuiFCVwgJWEX6mMUgHiB3qLux8QYDHXnbXAcgWxRy7S5JcnbMjdfWNSwFjpXnJVp2Fm5z
zvLVzcCqDLT2zC",
          "state_hash": "bs_2pAUexcNWE9HFruXUugY28yfUifWDh449JK1dDgdeMix5uk8Q",
          "time": 1543375246712,
```

```
          "transactions": {
            "th_2FHxDzpQMRTiRfpYRV3eCcsheHr1sjf9waxk7z6JDTVcgqZRXR": {
              "block_hash": "mh_ufiYLdN8am8fBxMnb6xq2K4MQKo4eFSCF5bgixq4EzKMtDUXP",
              "block_height": 1,
              "hash": "th_2FHxDzpQMRTiRfpYRV3eCcsheHr1sjf9waxk7z6JDTVcgqZRXR",
              "signatures":
"sg_Fipyxq5f3JS9CB3AQVCw1v9skqNBw1cdfe5W3h1t2MkviU19GQckERQZkqkaXWKowdTUvr7B1QbtWdH
jJHQcZApwVDdP9",
              "tx": {
                "amount": 150425,
                "fee": 101014,
                "nonce": 1,
                "payload": "790921-801018",
                "recipient_id":
"ak_26dopN3U2zgfJG4Ao4J4ZvLTf5mqr7WAgLAq6WxjxuSapZhQg5",
                "sender_id":
"ak_26dopN3U2zgfJG4Ao4J4ZvLTf5mqr7WAgLAq6WxjxuSapZhQg5",
                "type": "SpendTx",
                "version": 1
              }
            }
          },
          "txs_hash": "bx_8K5NtXK56QmUAsriAYocpqAUowJMsbEJmHEGrz7SRiu1g1yjo",
          "version": 1
        }
      },
      "miner": "ak_q9KDcpGHQ377rVS1TU2VSofby2tXWPjGvKizfGUC86gaq7rie",
      "nonce": "7537663592980547537",
      "pow": "[26922260, 37852188, 59020115, 60279463, 79991400, 85247410,
107259316, 109139865, 110742806, 135064096, 135147996, 168331414, 172261759,
199593922, 202230201, 203701465, 210434810, 231398482, 262809482, 271994744,
272584245, 287928914, 292169553, 362488698, 364101896, 364186805, 373099116,
398793711, 400070528, 409055423, 410928197, 423334086, 423561843, 428130074,
496454011, 501715005, 505858333, 514079183, 522053501, 526239399, 527666844,
532070334]",
      "prev_hash": "kh_pbtwgLrNu23k9PA6XCZnUbtsvEFeQGgavY4FS2do3QP8kcp2z",
      "prev_key_hash": "kh_pbtwgLrNu23k9PA6XCZnUbtsvEFeQGgavY4FS2do3QP8kcp2z",
      "state_hash": "bs_QDcwEF8e2DeetViw6ET65Nj1HfPrQh1uRkxtAsaGLntRGXpg7",
      "target": 522133279,
      "time": 1543373685748,
      "version": 1
    },
    "2": {
      "beneficiary": "ak_21rna3xrD7p32U3vpXPSmanjsnSGnh6BWFPC9Pe7pYxeAW8PpS",
      "hash": "kh_Z6iGf4ajdT5nhRMRtE7iLCii1BLQS4govtSiFwnfRtHZRxubz",
      "height": 2,
      "micro_blocks": {},
      "miner": "ak_2GRQehEg2PgyFKBhtfuGEBA5JR4JmQyKo2mxdT7kBcrKYKhE1i",
      "nonce": "5900433900970191660",
      "pow": "[5101167, 6731386, 32794521, 37862469, 82304394, 88947395, 96272418,
117165693, 128680663, 130957359, 138202691, 145997910, 148853998, 158275375,
161243335, 190430513, 198310271, 213658699, 216705056, 223898939, 235521909,
242195781, 244411339, 259255091, 274739784, 274765835, 298847001, 303666419,
```

```
308332831, 344614579, 352648945, 359486160, 364216435, 365891779, 371759238,
377325461, 379358071, 419687839, 439118573, 440188602, 479121064, 513335347]",
      "prev_hash": "mh_ufiYLdN8am8fBxMnb6xq2K4MQKo4eFSCF5bgixq4EzKMtDUXP",
      "prev_key_hash": "kh_29Gmo8RMdCD5aJ1UUrKd6Kx2c3tvHQu82HKsnVhbprmQnFy5bn",
      "state_hash": "bs_2pAUexcNWE9HFruXUugY28yfUifWDh449JK1dDgdeMix5uk8Q",
      "target": 522133279,
      "time": 1543375246777,
      "version": 1
    }
  },
  "total_micro_blocks": 1,
  "total_transactions": 1
}
```


`GET /middleware/height/at/<millis_since_epoch>`

What was the height at a certain time, measured in milliseconds since Jan 1 1970 (i.e. UNIX time multiplied by 1,000):

```
$ curl -s 'https://mdw.aepps.com/middleware/height/at/1543375246777' | jq
{
  "height": 2
}
```

`GET /middleware/names/active?<limit>&<page>`


```
$ curl -s 'https://mdw.aepps.com/middleware/names/active?limit=1' | jq
[
  {
    "id": 1486,
    "name": "o74.test",
    "name_hash": "nm_AhAqJKL8cPiBwip3RNKyGrvKookzr9pgkWps5ZPP8wijZXCXi",
    "created_at_height": 37015,
    "owner": "ak_Z7jX1axhYKUxCNeZCLhTazEaimCYibxUwuBveTjZfAskBCR7E",
    "expires_at": 87015,
    "pointers": [
      {
        "id": "ak_QSNbDgTDwn1izHWdJHM1EuMiMrmVya9sy7MA1pDtJYVCxFgT6",
        "key": "account_pubkey"
      }
    ]
  }
]
```


`GET /middleware/oracles/list?<limit>&<page>`

All oracles, most recently registered first.


```
$ curl -s 'https://mdw.aepps.com/middleware/oracles/list?limit=1' | jq
[
  {
    "block_height": 91132,
    "expires_at": 91932,
    "oracle_id": "ok_28QDg7fkF5qiKueSdUvUBtCYPJdmMEoS73CztzXCRAwMGKHKZh",
    "transaction_hash": "th_2eB18eh8tLM1GrqhkXAiqxocZvX9U4yrcJ9c1eRbfXqj6WTgYd",
    "tx": {
      "abi_version": 0,
      "account_id": "ak_28QDg7fkF5qiKueSdUvUBtCYPJdmMEoS73CztzXCRAwMGKHKZh",
```

```
      "fee": 20000000000000,
      "nonce": 751,
      "oracle_ttl": {
        "type": "delta",
        "value": 800
      },
      "query_fee": 1e+18,
      "query_format": "string",
      "response_format": "string",
      "ttl": 91142,
      "type": "OracleRegisterTx",
      "version": 1
    }
  }
]
```


`GET /middleware/oracles/<oracle_id>?<limit>&<page>`

This oracle's transactions:

```
$ curl -s
'https://mdw.aepps.com/middleware/oracles/ok_28QDg7fkF5qiKueSdUvUBtCYPJdmMEoS73Cztz
XCRAwMGKHKZh?limit=1' | jq
```


```
[
  {
    "query_id": "oq_Xp6fVk1uHBmTctRCwvaeeKRiBikN9fATdG8mo43TcRmPsYeGu",
    "request": {
      "fee": 900000,
      "nonce": 18560,
      "oracle_id": "ok_28QDg7fkF5qiKueSdUvUBtCYPJdmMEoS73CztzXCRAwMGKHKZh",
      "query": "aebtc",
      "query_fee": 500000,
      "query_ttl": {
        "type": "delta",
        "value": 100
      },
      "response_ttl": {
        "type": "delta",
        "value": 30
      },
      "sender_id": "ak_24jcHLTZQfsou7NvomRJ1hKEnjyNqbYSq2Az7DmyrAyUHPq8uR",
      "ttl": 44000,
      "type": "OracleQueryTx",
      "version": 1
    },
    "response": null
  }
]
```


`GET /middleware/reward/height/<height>`

The reward at a block height, which is comprised of the mining reward, and the fees for the transactions which are included

```
$ curl -s 'https://mdw.aepps.com/middleware/reward/height/10000' | jq
```


```
{
```

```
  "coinbase": "5831398157261209600",
  "fees": "10848",
  "height": 10000,
  "total": "5831398157261220448"
}
```

`GET /middleware/size/current`

The size of all transactions, in bytes, at the current height of the chain. This number is indicative.

$ curl -s 'https://mdw.aepps.com/middleware/size/current' | jq

{

"size": 499426140

}

`GET /middleware/size/height/<height>`

The same as above, but at some height

```
$ curl -s 'https://mdw.aepps.com/middleware/size/height/100' | jq
{
  "size": 234
}
```

`GET /middleware/status`

A status page, for monitoring purposes

```
$ curl -s 'https://mdw.aepps.com/middleware/status' | jq
{
  "OK": true,
  "queue_length": 0,
  "seconds_since_last_block": 73
}
```

The ‘OK’ field is set to false when the queue length is more than 2, and/or the seconds_since_last_block is > 1200 seconds. The teme can be overridden by the environment variable STATUS_MAX_BLOCK_AGE

`GET /middleware/transactions/account/<account>/count`

How many transactions does a particular account have?

```
$ curl -s
'https://mdw.aepps.com/middleware/transactions/account/ak_24jcHLTZQfsou7NvomRJ1hKEn
jyNqbYSq2Az7DmyrAyUHPq8uR/count' | jq
{
  "count": 19138
}
```

`GET /middleware/transactions/account/<sender>/to/<receiver>`

All SpendTX transactions from one account to another

```
$ curl -s
'https://mdw.aepps.com/middleware/transactions/account/ak_26dopN3U2zgfJG4Ao4J4ZvLTf
5mqr7WAgLAq6WxjxuSapZhQg5/to/ak_26dopN3U2zgfJG4Ao4J4ZvLTf5mqr7WAgLAq6WxjxuSapZhQg5'
| jq
{
  "transactions": [
    {
```

```
      "block_height": 1,
      "block_hash": "mh_ufiYLdN8am8fBxMnb6xq2K4MQKo4eFSCF5bgixq4EzKMtDUXP",
      "hash": "th_2FHxDzpQMRTiRfpYRV3eCcsheHr1sjf9waxk7z6JDTVcgqZRXR",
      "signatures": [

"sg_Fipyxq5f3JS9CB3AQVCw1v9skqNBw1cdfe5W3h1t2MkviU19GQckERQZkqkaXWKowdTUvr7B1QbtWdH
jJHQcZApwVDdP9"
      ],
      "tx": {
        "amount": 150425,
        "fee": 101014,
        "nonce": 1,
        "payload": "790921-801018",
        "recipient_id": "ak_26dopN3U2zgfJG4Ao4J4ZvLTf5mqr7WAgLAq6WxjxuSapZhQg5",
        "sender_id": "ak_26dopN3U2zgfJG4Ao4J4ZvLTf5mqr7WAgLAq6WxjxuSapZhQg5",
        "type": "SpendTx",
        "version": 1
      }
    }
  ]
}
```


`GET /middleware/transactions/account/<account>?<limit>&<page>`

All transactions for an account

```
$ curl -s
'https://mdw.aepps.com/middleware/transactions/account/ak_26dopN3U2zgfJG4Ao4J4ZvLTf
5mqr7WAgLAq6WxjxuSapZhQg5?limit=1' | jq
```


```
[
  {
    "block_hash": "mh_C2b6eKqXtgS1XTp4785228BZkMoA2M9SSFmggz7oi3i6xLMai",
    "block_height": 97924,
    "hash": "th_fZ6QPGAi8EGz6qnYC5nhCBCUBbBtAGn6DuaSLMRNyFMZqGFu7",
    "signatures": [

"sg_WeVByCNL7YuDwCC5G6PHuJuSjZaq9idx3vNVJGpyLT3e8C4ZW4hufqmzm7sJDLad4L5297913jqUKAu
PEZ4su3h3pVtHA"
    ],
    "time": 1561018747052,
    "tx": {
      "amount": 1e+18,
      "fee": 20000000000000,
      "nonce": 87,
      "payload": "ba_Xfbg4g==",
      "recipient_id": "ak_2vx4yNy2FUi7Fe2ZjKbKvpnabDTJE8RijtfAhQHNjY5zfj1We6",
      "sender_id": "ak_26dopN3U2zgfJG4Ao4J4ZvLTf5mqr7WAgLAq6WxjxuSapZhQg5",
      "type": "SpendTx",
      "version": 1
    }
  }
]
```


`GET /middleware/transactions/interval/<from>/<to>?<limit>&<page>`

All transactions between two heights (inclusive)

```
$ curl -s 'https://mdw.aepps.com/middleware/transactions/interval/1/3?limit=1' | jq
{
  "transactions": [
    {
      "block_height": 1,
      "block_hash": "mh_ufiYLdN8am8fBxMnb6xq2K4MQKo4eFSCF5bgixq4EzKMtDUXP",
      "hash": "th_2FHxDzpQMRTiRfpYRV3eCcsheHr1sjf9waxk7z6JDTVcgqZRXR",
      "signatures": [

"sg_Fipyxq5f3JS9CB3AQVCw1v9skqNBw1cdfe5W3h1t2MkviU19GQckERQZkqkaXWKowdTUvr7B1QbtWdH
jJHQcZApwVDdP9"
      ],
      "tx": {
        "amount": 150425,
        "fee": 101014,
        "nonce": 1,
        "payload": "790921-801018",
        "recipient_id": "ak_26dopN3U2zgfJG4Ao4J4ZvLTf5mqr7WAgLAq6WxjxuSapZhQg5",
        "sender_id": "ak_26dopN3U2zgfJG4Ao4J4ZvLTf5mqr7WAgLAq6WxjxuSapZhQg5",
        "type": "SpendTx",
        "version": 1
      }
    }
  ]
}
```

`GET /middleware/transactions/rate/<from>/<to>`

Returns the total of all transfers and the number of transactions for each date in a range

```
$ curl -s 'https://mdw.aepps.com/middleware/transactions/rate/20190101/20190105'
|jq
[
  {
    "amount": "460565852664889999406222",
    "count": 1886,
    "date": "2019-01-02"
  },
  {
    "amount": "479693021591472218661146",
    "count": 6805,
    "date": "2019-01-03"
  },
  {
    "amount": "511176955317249199450664",
    "count": 7301,
    "date": "2019-01-04"
  }
]
```

Websocket interface

The websocket interface, which listens by default on port 3020, gives asynchronous notifications when various events occur. You may subscribe to the generation of key blocks, micro blocks, and transactions, and you may also subscribe to any æternity type, i.e. anything that begins XX_ followed by a base58-encoded identifier. So for instance if you have an oracle ok_JcUaMCu9FzTwonCZkFE5BXsgxueagub9fVzywuQRDiCogTzse you may subscribe to this object and be notified of any events which relate to it--presumable you would be interested in queries, to which you would respond. Of course you can also subscribe to accounts, contracts, names, whatever you like.

The websocket interface accepts JSON-encoded commands to subscribe and unsubscribe, and answers these with the list of subscriptions. A session will look like this:

```
$ wscat -c ws://localhost:3022
connected (press CTRL+C to quit)
< connected
> {"op":"subscribe", "payload": "key_blocks"}
< ["key_blocks"]
<
{"payload":{"beneficiary":"ak_542o93BKHiANzqNaFj6UurrJuDuxU61zCGr9LJCwtTUg34kWt","h
ash":"kh_Ba9582bxExr7tsAfiWxtVUu5RNB621p3c3hNcZrFuYLkQ7ush","height":41838,"info":"
cb_Xfbg4g==","miner":"ak_2Y4ekjiUYS1J3XWLXGB2MuvpDC2zjB4i4JUpqCuuC9nWNNKnra","nonce
":55578800732524,"pow":[42398937,44145582,60062313,67268690,70386727,73311191,73976
863,80847263,85649902,91860174,95381587,117552688,120127730,120694109,124732915,130
547602,141123393,153513758,153717723,160456351,179135883,189963798,193935143,213809
942,228813186,229068838,253097583,253348228,293469539,294519175,353462944,361343692
,368029449,370749140,377762628,408045994,418987121,423343344,445325212,469339659,47
5159633,484059531],"prev_hash":"mh_2DwJimao7YZ4YBuBPfpjW1BfkNjb9SR14ALS7cmgEvnNzHh1
op","prev_key_hash":"kh_6Vfde6858MgjmD9cjkXy3HUd5XKUZMD1ZQFjqAc5KmaE6GwhE","state_h
ash":"bs_21sEUyDWVYJwigfNrE39HracXEW37Yz8oTAXyYh7qCBSnnUNXz","target":503757256,"ti
me":1550888063558,"version":1},"subscription":"key_blocks"}
> {"op":"subscribe", "payload": "micro_blocks"}
< ["key_blocks","micro_blocks"]
<
{"payload":{"hash":"mh_2Ut6zfHh9Sgn51betD6crogqncTRpNgFzLBpkaYxxQesZUtdhg","key_blo
ck_id":75308,"pof_hash":"no_fraud","prev_hash":"mh_2WwPA1XpE4sxcpZyfu6teuXDrZH3odGX
D3fStK8t32wt7XTX1v","prev_key_hash":"kh_Ba9582bxExr7tsAfiWxtVUu5RNB621p3c3hNcZrFuYL
kQ7ush","signature":"sg_CY7QDQou8XCH7Zc9vUPM1Zxqxu3F9Hsbx5p6zPKpkdqStdPC782bqH6KnF1
iwE81JhgK6Zm1dinNSXW1waRreLdWYiz2z","state_hash":"bs_tTFqiA3dmKz93SQRRuta1v5T71Y5YZ
2VuDH1HydZ7X1iX2b9b","time":1550888142120,"txs_hash":"bx_27TzDcsq2X5MybSsnUQEqAszLi
D1qpZY3f2TwUHYSKKwn4B9DM","version":1},"subscription":"micro_blocks"}
<
{"payload":{"hash":"mh_VZQBoajxRN9b9drJdTmpVqPnfhSrWWRuGdMh2Sm6W2g7v4y2M","key_bloc
k_id":75308,"pof_hash":"no_fraud","prev_hash":"mh_2Ut6zfHh9Sgn51betD6crogqncTRpNgFz
LBpkaYxxQesZUtdhg","prev_key_hash":"kh_Ba9582bxExr7tsAfiWxtVUu5RNB621p3c3hNcZrFuYLk
Q7ush","signature":"sg_RaaooJsVvfQwnMQVdWjNBB8NhNDAxbixh7db92SbfhEWNdn1cNrCMPyofRZn
oQMizcgygQByCphj8McH2EuPuDaVE9nMy","state_hash":"bs_yrrmRJKihvoFBWtLU8PLNeVGv82Ek8v
7mtsVqFqCRuCdcXVNS","time":1550888145120,"txs_hash":"bx_2uUunLXh3kzCuEBziFpvcd6Lp8H
YSPjV6TLFmCpfA9piXyvNkX","version":1},"subscription":"micro_blocks"}
```

As you can see, commands are sent as JSON, of the form:
{"op":"OPERATION", "payload": "PAYLOAD", target=”TARGET”}

where:
op can be subscribe or unsubscribe
payload can be ‘key_blocks’, ‘micro_blocks’, ‘transaction’ or ‘object’
target is only used when subscribing to objects, and can be any æternity object.

The response to a subscribe or unsubscribe is the set of current subscriptions.

Ways of running mdw

All-in-one

The simplest way to run mdw is with one process both populating the database and serving http requests. This uses the -s. -p and (maybe) -w options, for serve, populate and websocket. This method scales relatively well, its main disadvantage is that when the population process gets wedged (as it does relatively often, still), all services are down whilst the service restarts

Separate population and serving processes

http serving is a relatively stable feature, and so the simplest way to have a reliable service is currently to run one process populating the database (i.e. with the -p option) and another one or more with the -s option. **Important note:** counterintuitively, the -w (websocket) feature only works inside the **populate** process, not the (http) serving one(s). This may be corrected in future releases.

Monitoring

There is a python script, scripts/monitor.py which can be modified for your installation. It is intended to be run from cron, we run it like this:

```
0,5,10,15,20,25,30,35,40,45,50,55 * * * * /usr/bin/python3
/usr/local/bin/monitor.py
```

Service files

There are two provided in scripts, for loading, and http serving. Of course you can combine them for both at once.

Verification

The middleware has a -v option, which walks back through the chain, checking the DB version against the one that the node reports. You can use this to verify your database

Loading sets of blocks

the -H option can be used to load parts of the blockchain in isolation. Should you ever wish to load only a certain set of blocks, run mdw from the command-line, with arguments of these form:

-H300-400
-H1,2,3,4
-H1,2,3,6-9,100-200

and so on.