Infura 基础 API 功能

时间 / 版本	主要功能	备注
2022.01. 18	Infura 基础功能	Eth1: 统计
2022.01.19	Infura 基础功能	Eth2: 统计 ITX：部分统计 Filecoin：未统计 IPFS：未统计

编写：程华峥

1 Infura 简介

Infura 是 Web3 后端基础设施（Infrastructure as a service），给开发者提供 **API 服务**、开发所需 **Tools**。

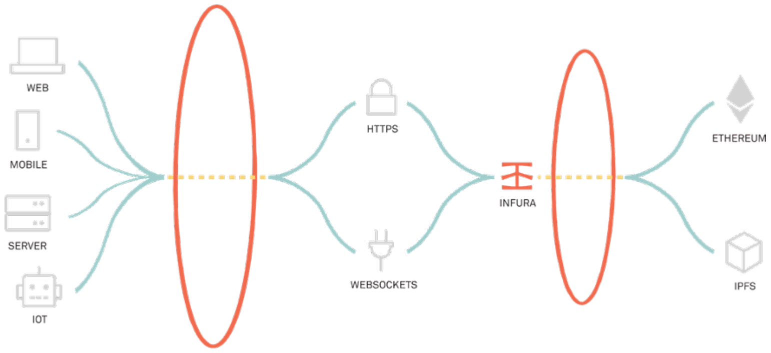
Infura 目前支持：以太坊 1.0、2.0（beta）、Transactions、Filecoin（beta）、IPFS（beta）。

Infura 通过分布式的云服务网络，减少了开发者开发运维中的**计算、存储**开销。

2 Why use an Infura API?

Infura API 可以带来 3 个好处：速度、存储、安全。相比项目方、应用方直接同步节点来说，直接通过 API 会更快获取到 Node 的状态。

Infura 的基础结构如下：



- 通过 HTTPS、Websocket，开发者可以直接获取 Eth、IPFS 的状态，相比用户自己搭建的 API 服务，Infura 的 API 可以提供 20 倍的接口访问速度[1]；
- Infura 提供 API 接口调用的监控、分析，同时提供 Eth 兼容的 Json-RPC，同时支持获取 Eth 的 Archive 节点数据；
- Infrua API 提供 99.9%的 uptime 和实时性[1]。

3 Infura 基础服务

3.1 Eth1.0

3.1.1 基础

* Authentication

利用 Project ID、Project Secret（需申请）增加接口调用的安全性和付费服务功能区分。

```
curl --user :[YOUR-PROJECT-SECRET] \
  https://<network>.infura.io/v3/YOUR-PROJECT-ID \
  -d '{"jsonrpc":"2.0","method":"eth_blockNumber","params":[],"id":1}'
```

利用 JWT 来增加接口安全性：

```
curl -H "Authorization: Bearer YOUR-JWT" \
  https://<network>.infura.io/v3/YOUR_PROOJECT_ID
```

* Choose a Network

Mainnet	JSON-RPC over HTTPS	https://mainnet.infura.io/v3/YOUR-PROJECT-ID

* JSON-RPC Methods

以太坊兼容的 Json-rpc 接口：

```
$ curl -X POST \
-H "Content-Type: application/json" \
--data '{"jsonrpc": "2.0", "id": 1, "method": "eth_blockNumber", "params": []}' \
"https://mainnet.infura.io/v3/YOUR-PROJECT-ID"
```

3.1.2 项目安全

* Authentication

Dashboard 里配置 Project secret required 和 JWT required。

* Request Rate Limiting

当 Project ID 泄露后，Dashboard 里可以配置 Per second requests rate limiting 参数，限制接口调用的频率。

Per day total requests 限制 00:00 UTC 之间的接口调用次数。

* Securing With Allowlists

应用环境不足以保护 Project Secret 和 Project ID 时，可以配置 allow list 来增加接口安全性。

* User-Agents

* HTTP Origin

* Method

使用 Project ID 指定接口的 method，即对应的 Project ID 可以限制调用的 API method。

```
    Allowlist entry: eth_call
    curl https://mainnet.infura.io/v3/YOUR-PROJECT-ID \
        -X POST \
        -H "Content-Type: application/json" \
        -d
'{"jsonrpc":"2.0","method":"eth_getBalance","params":["0xc94770007dda54cF92009BFF0dE90c06F603a09f",
"latest"],"id":1}'
    则接口调用 eth_call 不被允许。
```

* Securing With JWTs

3.1.3 Http / Websockets

Infura 支持 http(s)与 ws(s)两种 endpoint，兼容 json-rpc 的接口。

3.1.4 Archive data

Eth 节点的 Full mode 不需要维持所有的节点状态，只需要保持最近的状态(128 个 Block 的状态)即可完成状态验证 / 状态重组[2]。Archive mode 是保留历史所有状态，相应所需的存储也会更大。

3.1.5 Network Add-Ons

* Arbitrum

* Optimistic Ethereum

* Polygon PoS

3.1.6 Error Codes

CODE	MESSAGE	MEANING	CATEGORY
-32700	Parse error	Invalid JSON	standard
-32600	Invalid request	JSON is not a valid request object	standard
-32601	Method not found	Method does not exist	standard
-32602	Invalid params	Invalid method parameters	standard
-32603	Internal error	Internal JSON-RPC error	standard
-32000	Invalid input	Missing or invalid parameters	non-standard
-32001	Resource not found	Requested resource not found	non-standard
-32002	Resource unavailable	Requested resource not available	non-standard
-32003	Transaction rejected	Transaction creation failed	non-standard
-32004	Method not supported	Method is not implemented	non-standard
-32005	Limit exceeded	Request exceeds defined limit	non-standard
-32006	JSON-RPC version not supported	Version of JSON-RPC protocol is not supported	non-standard

一个参考 error：

```
{
    "id": 1337
    "jsonrpc": "2.0",
    "error": {
        "code": -32003,
        "message": "Transaction rejected"
    }
}
```

3.1.7 Value encoding

* Quantity

所有的数量相关遵循规则：

a) Hex-encoded

b) "0x"-prefixed

c) zero as "0x0"

* Block Identifier

需要 block identifier 的接口如下：

- eth_getBalance
- eth_getStorageAt
- eth_getTransactionCount
- eth_getCode
- eth_call
- eth_getProof

The block identifier is a JSON object with the following fields:

PROPERTY	TYPE	DESCRIPTION
[blockNumber]	{Quantity}	The block in the canonical chain with this number
OR [blockHash]	{Data}	The block uniquely identified by this hash. The blockNumber and blockHash properties are mutually exclusive; exactly one of them must be set.
requireCanonical	{boolean}	(optional) Whether or not to throw an error if the block is not in the canonical chain as described below. Only allowed in conjunction with the blockHash tag. Defaults to false.

* Data

a) hex-encoded

b) "0x"-prefixed

c) expressed using two hex digits per byte（保证字节数据为偶数）

3.1.8 Transactions

Infura Transactions (ITX)，支持转发 eth 的交易到节点。

a) 使用 etherjs 的 sdk，调用 Infura Provider；

b) signer.sendTransaction

c) signer.signMessage

d) Call a contract method

e) "relay_sendTransaction"

f) "relay_getTransactionStatus"

g) "relay_getBalance"

3.1.9 other methods

* eth_subscribe

* eth_unsubscribe

* eth_blockNumber …

* eth_getFilterChanges

* eth_getFilterLogs

* eth_newBlockFilter

* eth_newFilter

* eth_newPendingTransactionFilter

3.2 Eth2

基础的 Authentication、Rate Limiting、参考 Eth1。

主要介绍 Eth2 特有接口：

3.2.1 Beacon

接口	说明	备注
Get chain genesis details	Retrieve details of the chain's genesis which can be used to identify chain.	GET
Get state root	Calculates HashTreeRoot for state with given 'state_id'. If stateId is root, same value will be returned.	

Get state fork	Returns Fork object for state with given 'state_id'.	
Get state finality checkpoints	Returns finality checkpoints for state with given 'state_id'. In case finality is not yet achieved, checkpoint should return epoch 0 and ZERO_HASH as root.	
Get validators from state	Returns filterable list of validators with their balance, status and index.	
Get validator from state	Returns validator specified by state and id or public key along with status and balance.	
Get validator balances from state	Returns filterable list of validator balances.	
Get committees at state	Retrieves the committees for the given state.	
Get block headers	Retrieves block headers matching given query. By default it will fetch current head slot blocks.	
Get block header	Retrieves block header for given block id.	
Publish a signed block	Submit a signed beacon block to the beacon node to be imported. The beacon node performs the required validation.	POST
Get block	Retrieves block details for given block id.	
Get block root	Retrieves hashTreeRoot of BeaconBlock/BeaconBlockHeader	
Get block attestations	Retrieves attestations included in requested block.	
Get attestations	Retrieves attestations known by the node but not necessarily incorporated into any block.	
Submit signed attestations	Submit signed attestations to the beacon node to be validated and submitted if valid. This endpoint does not protected against slashing.	POST
Get AttesterSlashings	Retrieves attester slashings known by the node but not necessarily incorporated into any block.	
Submit attester slashing object	Submits attester slashing object to node's pool and if passes validation node MUST broadcast it to network.	POST
Get proposer slashings	Retrieves proposer slashings known by the node but not necessarily incorporated into any block.	
Submit proposer slashing object	Submits proposer slashing object to node's pool and if passes validation node MUST broadcast it to network.	POST
Get signed voluntary exits	Retrieves voluntary exits known by the node but not necessarily incorporated into any block.	
Submit signed voluntary exit	Submits signed voluntary exit object to node's pool and if it passes validation node MUST broadcast it to network.	POST

3.2.2 Validator Required Api

Submit signed attestations	Submit signed attestations to the beacon node to be validated and submitted if valid. This endpoint does not protected against slashing.	POST
Subscribe to node events	Provides endpoint to subscribe to beacon node Server-Sent-Events stream.	POST
Get node syncing status	Requests the beacon node to describe if it's currently syncing or not, and if it is, what block it is up to.	

Get attester duties	Requests the beacon node to provide a set of attestation duties, which should be performed by validators, for a particular epoch. Duties should only need to be checked once per epoch, however a chain reorganization (of > MIN_SEED_LOOKAHEAD epochs) could occur, resulting in a change of duties.	
Get proposer duties	Request beacon node to provide all validators that are scheduled to propose a block in the given epoch.	
Produce an AttestationData	Returns attestation data for the block at the specified non-finalized slot.	
Get aggregated attestations	Aggregates all attestations matching given attestation data root and slot.	
Publish aggregate and proofs	Verifies given aggregate and proofs and publishes it on appropriate gossipsub topic.	POST
Subscribe to a committee subnet	After Beacon node receives this request, search using discv5 for peers related to this subnet and replace current peers with those ones if necessary If validator is_aggregator, beacon node must: * announce subnet topic subscription on gossipsub * aggregate attestations received on that subnet	POST
Get spec params	Retrieve specification configuration used on this node.	

3.2.3 Node

3.2.4 Validator

3.2.5 Config

3.3 Transactions

Infura Transactions (ITX)是 Infura 在 Eth 上部署的支持 relay 交易的合约，主要实现了 3 个接口：

* relay_sendTransaction

* relay_getTransactionStatus

* relay_getBalance

3.4 Filecoin

3.5 IPFS