eth-typing Documentation

Release 4.2.3

The Ethereum Foundation

May 14, 2024

CONTENTS

Common type annotations for ethereum python packages.

CHAPTER

ONE

CONTENTS

1.1 Types

The following types are available from the `eth_typing` module.

i.e.

```
from eth_typing import TypeStr
```

1.1.1 Application Binary Interface

`eth_typing.abi.`**`ABI`**

List of components representing function and event interfaces (elements of an ABI).

alias of `Sequence`[`Union`[*`ABIFunction`*, *`ABIConstructor`*, *`ABIFallback`*, *`ABIReceive`*, *`ABIEvent`*, *`ABIError`*]]

`class eth_typing.abi.`**`ABIComponent`**`(*args, **kwargs)`

TypedDict representing an *ABIElement* component.

`components:` `Sequence[`*`ABIComponent`*`]`

List of nested *ABI* components for ABI types.

`name:` `str`

Name of the component.

`type:` `str`

Type of the component.

`class eth_typing.abi.`**`ABIConstructor`**`(*args, **kwargs)`

TypedDict representing the *ABI* for a constructor function.

`constant:` `bool`

`inputs:` `Sequence[`*`ABIComponent`*`]`

Function input components.

`payable:` `bool`

`stateMutability:` `Literal['pure', 'view', 'nonpayable', 'payable']`

`type:` `Literal['constructor']`

Type of the constructor function.

eth_typing.abi.**ABIElement**

Base type for *ABIFunction* and *ABIEvent* types.

alias of Union[*ABIFunction*, *ABIConstructor*, *ABIFallback*, *ABIReceive*, *ABIEvent*, *ABIError*]

class eth_typing.abi.**ABIError**(**args*, ***kwargs*)

TypedDict representing the *ABI* for an error.

inputs: Sequence[*ABIComponent*]

Error input components.

name: str

Name of the error.

type: Literal['error']

Type of the error.

class eth_typing.abi.**ABIEvent**(**args*, ***kwargs*)

TypedDict to represent the *ABI* for an event.

anonymous: bool

If True, event is anonymous. Cannot filter the event by name.

inputs: Sequence[*ABIComponent*]

Input components for the event.

name: str

Event name identifier.

type: Literal['event']

Event ABI type.

class eth_typing.abi.**ABIFallback**(**args*, ***kwargs*)

TypedDict representing the *ABI* for a fallback function.

constant: bool

payable: bool

stateMutability: Literal['pure', 'view', 'nonpayable', 'payable']

type: Literal['fallback']

Type of the fallback function.

class eth_typing.abi.**ABIFunction**(**args*, ***kwargs*)

TypedDict representing the *ABI* for a function.

constant: bool

inputs: Sequence[*ABIComponent*]

Function input components.

name: str

Name of the function.

outputs: Sequence[*ABIComponent*]

Function output components.

payable: bool

stateMutability: Literal['pure', 'view', 'nonpayable', 'payable']

type: Literal['function']

Type of the function.

class eth_typing.abi.**ABIFunctionInfo**(**args*, ***kwargs*)

TypedDict to represent an *ABIFunction* with the function selector and corresponding arguments.

abi: *ABIFunction*

ABI for the function interface.

arguments: Tuple[Any, ...]

Function input components.

selector: HexStr

Solidity Function selector sighash.

class eth_typing.abi.**ABIFunctionType**(**args*, ***kwargs*)

TypedDict representing the *ABI* for all function types.

This is the base type for functions. Please use ABIFunction, ABIConstructor, ABIFallback or ABIReceive instead.

constant: bool

Function is constant and does not change state. Deprecated in favor of stateMutability pure and view.

payable: bool

Contract is payable to receive ether on deployment. Deprecated in favor of stateMutability payable and nonpayable.

stateMutability: Literal['pure', 'view', 'nonpayable', 'payable']

State mutability of the constructor.

class eth_typing.abi.**ABIReceive**(**args*, ***kwargs*)

TypedDict representing the *ABI* for a receive function.

constant: bool

payable: bool

stateMutability: Literal['pure', 'view', 'nonpayable', 'payable']

type: Literal['receive']

Type of the receive function.

eth_typing.abi.**Decodable**

Binary data to be decoded.

alias of Union[bytes, bytearray]

eth_typing.abi.**TypeStr**

String representation of a data type.

1.1.2 Enumerables

ForkName

Class that contains the different names used to represent hard forks on the Ethereum network.

```
class ForkName:
    Frontier = 'Frontier'
    Homestead = 'Homestead'
    EIP150 = 'EIP150'
    EIP158 = 'EIP158'
    Byzantium = 'Byzantium'
    Constantinople = 'Constantinople'
    Metropolis = 'Metropolis'
```

ChainId

IntEnum class defining EVM-compatible network name enums as their respective chain id *int* values.

To learn more about chain ids, see CAIP-2 for details.

The list of chain ids is available from the ethereum-lists/chains repository.

```
class ChainId(IntEnum):
    # L1 networks
    ETH = 1
    EXP = 2
    ROP = 3
    RIN = 4
    GOR = 5
    # L2 networks
    OETH = 10
    GNO = 100
```

1.1.3 Discovery

NodeID

A 32-byte identifier for a node in the Discovery DHT

```
NodeID = NewType('NodeID', bytes)
```

1.1.4 EthPM

ContractName

Any string conforming to the regular expression `[a-zA-Z][a-zA-Z0-9_]{0,255}`.

```
ContractName = NewType('ContractName', str)
```

URI

Any string that represents a URI.

```
URI = NewType('URI', str)
```

1.1.5 EVM

Address

Any bytestring representing a canonical address.

```
Address = NewType('Address', bytes)
```

HexAddress

Any *HexStr* representing a hex encoded address.

```
HexAddress = NewType('HexAddress', HexStr)
```

ChecksumAddress

Any *HexAddress* that is formatted according to ERC55.

```
ChecksumAddress = NewType('ChecksumAddress', HexAddress)
```

AnyAddress

Any of *Address*, *HexAddress*, *ChecksumAddress*.

```
AnyAddress = TypeVar('AnyAddress', Address, HexAddress, ChecksumAddress)
```

Hash32

Any 32 byte hash.

```
Hash32 = NewType('Hash32', bytes)
```

BlockNumber

Any integer that represents a valid block number on a chain.

```
BlockNumber = NewType('BlockNumber', int)
```

BlockIdentifier

Either a 32 byte hash or an integer block number

```
BlockIdentifier = Union[Hash32, BlockNumber]
```

1.1.6 Encodings

HexStr

Any string that is hex encoded.

```
HexStr = NewType('HexStr', str)
```

Primitives

Any of *bytes*, *int*, or *bool* used as the *Primitive* arg for conversion utils in ETH-Utils.

```
Primitives = Union[bytes, int, bool]
```

1.2 Release Notes

1.2.1 eth-typing v4.2.3 (2024-05-06)

Features

- Update networks types with the latest. (#72)

1.2.2 eth-typing v4.2.2 (2024-04-29)

Bugfixes

- Fixes types that were incorrectly defined for `ABI` utils. (#62)

Features

- Update network type mappings. (#70)

Miscellaneous Changes

- #68

1.2.3 eth-typing v4.2.1 (2024-04-16)

Bugfixes

- Put back types used for *EthPM*: *ContractName*, *Manifest*, and *URI*. (#64)

1.2.4 eth-typing v4.2.0 (2024-04-15)

Features

- Add type definitions to represent contract `ABI` s. (#61)

Removals

- Remove types related to the EthPM module which has been removed from `web3.py` (#60)

1.2.5 eth-typing v4.1.0 (2024-04-01)

Features

- Add python3.12 support (#57)

Internal Changes - for eth-typing Contributors

- Merge template updates, adding build tests for all docs formats, add `blocklint` to lint tools (#57)

1.2.6 eth-typing v4.0.0 (2024-01-09)

Breaking changes

- Drop python 3.7 support (#55)

Internal Changes - for eth-typing Contributors

- Merge updates from the project template, notably: use `pre-commit` for linting and change the name of the `master` branch to `main` (#55)
- Fixed booleans in `pyproject.toml` and added a test for the presence of the `eth_typing.__version__` attribute (#56)

1.2.7 eth-typing v3.5.2 (2023-11-07)

Miscellaneous Changes

- #54

1.2.8 eth-typing v3.5.1 (2023-10-20)

Internal Changes - for eth-typing Contributors

- Add script to maintain Network constants listed in the networks module. (#51)
- Add `types-setuptools` to support pkg_resources and __version__ (#52)

1.2.9 eth-typing v3.5.0 (2023-09-29)

Features

- Borrowing from the typing in web3.py, open up `BlockIdentifier` to include `BlockParams` (e.g. “latest”, “finalized”, etc..) as well as other valid values. (#47)
- Add an `IntEnum` class, `ChainId`, defining EVM-compatible network name enums as their respective chain id `int` values. (#49)

Internal Changes - for eth-typing Contributors

- Add the tests/ directory to the distributed tarball (#46)
- Added `build.os` config for readthedocs (#48)
- Fix release command by checking the git remote upstream configuration and merge other minor template updates. (#50)

1.2.10 eth-typing v3.4.0 (2023-06-07)

Improved Documentation

- pull in ethereum-python-project-template updates (#44)

Features

- Add `Cancun` to `ForkName` enum. (#45)

Internal Changes - for eth-typing Contributors

- remove unused docs deps, bump version of remaining (#43)
- pull in ethereum-python-project-template updates (#44)
- For CircleCI builds, update `pip` and pip install `tox` under sys instead of `--user` to avoid `virtualenv` versioning issues. (#45)

1.2.11 v3.3.0 (2023-03-08)

Features

- Add `Shanghai` to `ForkName` enum. (#39)
- Add support for python `3.11`. (#40)

Internal Changes - for eth-typing Contributors

- `tox` related updates for `make docs` to work properly. Remove some old references to python `3.5` and `3.6`. (#39)
- Bump `mypy` version to `0.910` to avoid issues installing the "[dev]" extra on Python 3.10. Update test suite to require installing the full dependency suite to help catch these errors. (#41)

1.2.12 v3.2.0 (2022-09-14)

Features

- Add `Merge` to `ForkName` enum (#34)

Bugfixes

- Pin Python version to <4 instead of <3.11 (#37)
- Rename `Merge` to `Paris` in `ForkNameEnum` (#38)

1.2.13 v3.1.0 (2022-06-22)

Features

- Setup towncrier to generate release notes from fragment files to ensure a higher standard for release notes. (#16)
- Add new BLSPrivateKey type for BLS private key (#23)
- Add __all__ property to __init__.py with appropriate types to explicitly export (#28)
- Add GrayGlacier to ForkName enum (#30)

Miscellaneous changes

- #32

1.2.14 v3.0.0 (2021-11-15)

- Update ForkName enum to include Berlin, London, and ArrowGlacier
- Update Python support to include python 3.8-3.10
- Remove Python 3.5 support

1.2.15 v2.2.0 (2019-10-31)

- Update ForkName enum to include ConstantinopleFix and Istanbul

1.2.16 v2.1.0 (2019-10-31)

- Add BLS types

1.2.17 v2.0.0 (2019-10-31)

- Expose Type Hints as per PEP 561

1.2.18 v1.0.0 (2018-06-08)

- Added annotations from py-evm.

1.2.19 v0.3.1 (2018-06-07)

- Removed eth-utils requirement.

1.2.20 v0.3.0 (2018-06-07)

- Updated `eth-utils` requirement.

1.2.21 v0.2.0 (2018-06-07)

- Launched repository, claimed names for pip, RTD, github, etc.

CHAPTER TWO

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

INDEX