Idena Client

Dec 02, 2020

Contents:

1 API Reference **3**
- 1.1 `idena` . . . 3

2 Installation **19**

3 Using Idena **21**
- 3.1 Calling *client.init* . . . 21
- 3.2 Setting environment variables . . . 21

4 Getting Blockchain Info **23**

5 Indices and tables **25**

Python Module Index **27**

Index **29**

This is Python wrapper for Idene Node.

Please checkout API refrence for avaliable APIs.

CHAPTER 1

API Reference

This page contains auto-generated API reference documentation[1].

1.1 idena

1.1.1 Subpackages

idena.apis

Submodules

idena.apis.account

Module Contents

Functions

create_account(password: None) → str
get_accounts() → List[str]
lock_account(address: str) → None
unlock_account(address: str, password: str = None, time: int = None) → None

idena.apis.account.**create_account**(*password: None*) → str

idena.apis.account.**get_accounts**() → List[str]

idena.apis.account.**lock_account**(*address: str*) → None

[1] Created with sphinx-autoapi

idena.apis.account.**unlock_account**(*address: str*, *password: str = None*, *time: int = None*) → None

idena.apis.blockchain

Module Contents

Functions

check_sync() → types.Sync	
get_address_pending_transactions(address: str) → types.AddressTransactions	
get_address_transactions(address: str, count: int = 5, token: str = None) → types.AddressTransactions	
get_block_by_hash(hash: str) → types.Block	
get_block_by_height(height: int) → types.Block	
get_burnt_coins() → Optional[types.BurntCoins]	
get_fee_rate() → decimal.Decimal	
get_last_block() → types.Block	
get_mempool() → Optional[str]	
get_raw_tx(nonce: int = None, epoch: int = None, from_: str = None, to: str = None, amount: decimal.Decimal = None, maxFee: decimal.Decimal = None, payload: str = None, tips: decimal.Decimal = None, useProto: bool = None, type: types.TxType = types.TxType.SendTx) → str	
get_transaction(hash: str) → types.Transaction	
get_transaction_receipt(hash: str) → Optional[types.TxReceipt]	
send_raw_tx(tx: str) → str	

idena.apis.blockchain.**check_sync**() → types.Sync

idena.apis.blockchain.**get_address_pending_transactions**(*address: str*) → types.AddressTransactions

idena.apis.blockchain.**get_address_transactions**(*address: str*, *count: int = 5*, *token: str = None*) → types.AddressTransactions

idena.apis.blockchain.**get_block_by_hash**(*hash: str*) → types.Block

idena.apis.blockchain.**get_block_by_height**(*height: int*) → types.Block

idena.apis.blockchain.**get_burnt_coins**() → Optional[types.BurntCoins]

idena.apis.blockchain.**get_fee_rate**() → decimal.Decimal

idena.apis.blockchain.**get_last_block**() → types.Block

idena.apis.blockchain.**get_mempool**() → Optional[str]

idena.apis.blockchain.**get_raw_tx**(*nonce: int = None*, *epoch: int = None*, *from_: str = None*, *to: str = None*, *amount: decimal.Decimal = None*, *maxFee: decimal.Decimal = None*, *payload: str = None*, *tips: decimal.Decimal = None*, *useProto: bool = None*, *type: types.TxType = types.TxType.SendTx*) → str

idena.apis.blockchain.**get_transaction**(*hash: str*) → types.Transaction

idena.apis.blockchain.**get_transaction_receipt**(*hash: str*) → Optional[types.TxReceipt]

idena.apis.blockchain.**send_raw_tx**(*tx: str*) → str

idena.apis.contract

Module Contents

Functions

call_contract(from_: str = None, contract: str = None, method: str = None, amount: decimal.Decimal = None, maxFee: decimal.Decimal = None, args: List[types.DynamicArg] = None, broadcastBlock: int = None) → str	
deploy_contract(from_: str = None, codeHash: str = None, amount: decimal.Decimal = None, maxFee: decimal.Decimal = None, args: List[types.DynamicArg] = None) → str	
estimate_call(from_: str = None, contract: str = None, method: str = None, amount: decimal.Decimal = None, maxFee: decimal.Decimal = None, args: List[types.DynamicArg] = None, broadcastBlock: int = None) → types.ContractTxReceipt	
estimate_deploy(from_: str = None, codeHash: str = None, amount: decimal.Decimal = None, maxFee: decimal.Decimal = None, args: List[types.DynamicArg] = None) → types.ContractTxReceipt	
estimate_terminate(from_: str = None, contract: str = None, maxFee: decimal.Decimal = None, args: List[types.DynamicArg] = None) → types.ContractTxReceipt	
get_stake(contract: str) → types.Stake	
read_contract_data(contract: str = None, key: str = None, format: str = None) → Any	
read_events(contract: str = None) → None	
readonly_call_contract(contract: str = None, method: str = None, format: str = None, args: List[types.DynamicArg] = None) → Any	
subscribe_to_contract_event(contract: str = None, event: str = None) → None	

Continued on next page

Table 3 – continued from previous page

terminate_contract(from_: str = None, contract: str = None, maxFee: decimal.Decimal = None, args: List[types.DynamicArg] = None) → str	
unsubscribe_from_contract_event(contract: str = None, event: str = None) → None	

idena.apis.contract.**call_contract**(*from_: str = None*, *contract: str = None*, *method: str = None*, *amount: decimal.Decimal = None*, *maxFee: decimal.Decimal = None*, *args: List[types.DynamicArg] = None*, *broadcastBlock: int = None*) → str

idena.apis.contract.**deploy_contract**(*from_: str = None*, *codeHash: str = None*, *amount: decimal.Decimal = None*, *maxFee: decimal.Decimal = None*, *args: List[types.DynamicArg] = None*) → str

idena.apis.contract.**estimate_call**(*from_: str = None*, *contract: str = None*, *method: str = None*, *amount: decimal.Decimal = None*, *maxFee: decimal.Decimal = None*, *args: List[types.DynamicArg] = None*, *broadcastBlock: int = None*) → types.ContractTxReceipt

idena.apis.contract.**estimate_deploy**(*from_: str = None*, *codeHash: str = None*, *amount: decimal.Decimal = None*, *maxFee: decimal.Decimal = None*, *args: List[types.DynamicArg] = None*) → types.ContractTxReceipt

idena.apis.contract.**estimate_terminate**(*from_: str = None*, *contract: str = None*, *maxFee: decimal.Decimal = None*, *args: List[types.DynamicArg] = None*) → types.ContractTxReceipt

idena.apis.contract.**get_stake**(*contract: str*) → types.Stake

idena.apis.contract.**read_contract_data**(*contract: str = None*, *key: str = None*, *format: str = None*) → Any

idena.apis.contract.**read_events**(*contract: str = None*) → None

idena.apis.contract.**readonly_call_contract**(*contract: str = None*, *method: str = None*, *format: str = None*, *args: List[types.DynamicArg] = None*) → Any

idena.apis.contract.**subscribe_to_contract_event**(*contract: str = None*, *event: str = None*) → None

idena.apis.contract.**terminate_contract**(*from_: str = None*, *contract: str = None*, *maxFee: decimal.Decimal = None*, *args: List[types.DynamicArg] = None*) → str

idena.apis.contract.**unsubscribe_from_contract_event**(*contract: str = None*, *event: str = None*) → None

idena.apis.dna

Module Contents

Functions

activate_invite(nonce: int = None, epoch: int = None, key: str = None) → types.Invite
activate_invite_to_random_address(nonce: int = None, epoch: int = None, key: str = None) → types.ActivateInviteToRandomAddr
become_offline(nonce: int = None, epoch: int = None) → str
become_online(nonce: int = None, epoch: int = None) → str
burn(nonce: int = None, epoch: int = None, from_: str = None, amount: decimal.Decimal = None, maxFee: decimal.Decimal = None, key: str = None) → str
change_profile(info: str = None, nickname: str = None, maxFee: decimal.Decimal = None) → types.ChangeProfile
export_key(password: str) → str
get_balance(address: str) → types.Balance
get_ceremony_intervals() → types.CeremonyIntervals
get_coinbase_address() → str
get_current_process() → types.State
get_epoch() → types.Epoch
get_identities() → List[types.Identity]
get_identity(address: str) → types.Identity
get_profile(address: str = None) → types.Profile
isValidationReady() → bool
send_invite(nonce: int = None, epoch: int = None, to: str = None, amount: decimal.Decimal = None) → types.Invite
send_tx(nonce: int = None, epoch: int = None, from_: str = None, to: str = None, amount: decimal.Decimal = None, maxFee: decimal.Decimal = None, payload: str = None, tips: decimal.Decimal = None, useProto: bool = None, type: types.TxType = types.TxType.SendTx) → str

idena.apis.dna.**change_god_address**

idena.apis.dna.**kill_identity**

idena.apis.dna.**kill_invitee**

idena.apis.dna.**send_dna**

idena.apis.dna.**activate_invite**(*nonce: int = None*, *epoch: int = None*, *key: str = None*) → types.Invite

idena.apis.dna.**activate_invite_to_random_address**(*nonce: int = None*, *epoch: int = None*, *key: str = None*) → types.ActivateInviteToRandomAddr

idena.apis.dna.**become_offline**(*nonce: int = None*, *epoch: int = None*) → str

idena.apis.dna.**become_online**(*nonce: int = None*, *epoch: int = None*) → str

idena.apis.dna.**burn**(*nonce: int = None*, *epoch: int = None*, *from_: str = None*, *amount: decimal.Decimal = None*, *maxFee: decimal.Decimal = None*, *key: str = None*) → str

idena.apis.dna.**change_profile**(*info: str = None*, *nickname: str = None*, *maxFee: decimal.Decimal = None*) → types.ChangeProfile

idena.apis.dna.**export_key**(*password: str*) → str

idena.apis.dna.**get_balance**(*address: str*) → types.Balance

idena.apis.dna.**get_ceremony_intervals**() → types.CeremonyIntervals

idena.apis.dna.**get_coinbase_address**() → str

idena.apis.dna.**get_current_process**() → types.State

idena.apis.dna.**get_epoch**() → types.Epoch

idena.apis.dna.**get_identities**() → List[types.Identity]

idena.apis.dna.**get_identity**(*address: str*) → types.Identity

idena.apis.dna.**get_profile**(*address: str = None*) → types.Profile

idena.apis.dna.**isValidationReady**() → bool

idena.apis.dna.**send_invite**(*nonce: int = None*, *epoch: int = None*, *to: str = None*, *amount: decimal.Decimal = None*) → types.Invite

idena.apis.dna.**send_tx**(*nonce: int = None*, *epoch: int = None*, *from_: str = None*, *to: str = None*, *amount: decimal.Decimal = None*, *maxFee: decimal.Decimal = None*, *payload: str = None*, *tips: decimal.Decimal = None*, *useProto: bool = None*, *type: types.TxType = types.TxType.SendTx*) → str

idena.apis.flip

Module Contents

Functions

delete_flip(hash: str) → str	
flip_words(hash: str) → Tuple[int, int]	
get_flip(hash: str) → types.Flip	
get_long_flip_hashes() → List[types.FlipHashes]	
get_raw_flip(hash: str) → types.RawFlip	
get_short_flip_hashes() → List[types.FlipHashes]	
submit_flip(publicHex: str = None, privateHex: str = None, pairId: str = None, hex: str = None) → types.FlipSubmit	
submit_long_answers(answers: types.FlipAnswers) → str	
submit_short_answers(answers: types.FlipAnswers) → str	

idena.apis.flip.**delete_flip**(*hash: str*) → str

idena.apis.flip.**flip_words**(*hash: str*) → Tuple[int, int]

idena.apis.flip.**get_flip**(*hash: str*) → types.Flip

idena.apis.flip.**get_long_flip_hashes**() → List[types.FlipHashes]

idena.apis.flip.**get_raw_flip**(*hash: str*) → types.RawFlip

idena.apis.flip.**get_short_flip_hashes**() → List[types.FlipHashes]

idena.apis.flip.**submit_flip**(*publicHex: str = None*, *privateHex: str = None*, *pairId: str = None*, *hex: str = None*) → types.FlipSubmit

idena.apis.flip.**submit_long_answers**(*answers: types.FlipAnswers*) → str

idena.apis.flip.**submit_short_answers**(*answers: types.FlipAnswers*) → str

idena.apis.net

Module Contents

Functions

add_peers(url) → str	
get_ipfs_addr() → str	
get_peers() → List[types.Peer]	

idena.apis.net.**add_peers**(*url*) → str

idena.apis.net.**get_ipfs_addr**() → str

idena.apis.net.**get_peers**() → List[types.Peer]

1.1.2 Submodules

idena.client

Module Contents

Functions

init(rpc_node: str, api_key: str) → None	Initialization

idena.client.**init**(*rpc_node: str*, *api_key: str*) → None

Initialization

Args: rpc_node (str): URL of rpc node api_key (str): API-Key

idena.exceptions

Module Contents

exception idena.exceptions.**IdenaException**(*code*, *message*)
Bases: Exception

Common base class for all non-exit exceptions.

__repr__(*self*)
Return repr(self).

idena.types

Module Contents

Classes

ActivateInviteToRandomAddr	
AddressTransactions	
Answer	
Balance	
BaseTxArgs	
Block	
BurntCoins	
CeremonyIntervals	
ChangeProfile	
ContractTxReceipt	
DynamicArg	Typed version of namedtuple.
Epoch	
Flip	
FlipAnswer	Typed version of namedtuple.
FlipAnswers	Typed version of namedtuple.
FlipHashes	
FlipWords	
Grade	
Identity	
Invite	
Peer	Typed version of namedtuple.
Profile	
RawFlip	
SendTxArgs	
Stake	Typed version of namedtuple.
State	
Sync	
Transaction	
TxAddr	
TxEvent	
TxReceipt	
TxType	

idena.types.**FlipSubmit**

class idena.types.**ActivateInviteToRandomAddr**

```
    address :str
    hash :str
    key :str
class idena.types.AddressTransactions

    token :Optional[str]
    transactions :Optional[List[Transaction]]
class idena.types.Answer

    left = 1
    none = 0
    right = 2
class idena.types.Balance

    balance :decimal.Decimal
    nonce :int
    stake :decimal.Decimal
class idena.types.BaseTxArgs

    epoch :int
    nonce :int
class idena.types.Block

    coinbase :str
    flags :List[str]
    hash :str
    height :int
    identityRoot :str
    ipfsCid :str
    isEmpty :bool
    offlineAddress :str
    parentHash :str
    root :str
    timestamp :int
    transactions :List[str]
class idena.types.BurntCoins
```

address :str

amount :decimal.Decimal

key :str

class idena.types.**CeremonyIntervals**

FlipLotteryDuration :float

LongSessionDuration :float

ShortSessionDuration :float

class idena.types.**ChangeProfile**

hash :str

txHash :str

class idena.types.**ContractTxReceipt**

contract :str

error :str

gasCost :decimal.Decimal

gasUsed :int

success :bool

txFee :decimal.Decimal

txHash :str

class idena.types.**DynamicArg**

Bases: typing.NamedTuple

Typed version of namedtuple.

Usage in Python versions >= 3.6:

```
class Employee(NamedTuple):
    name: str
    id: int
```

This is equivalent to:

```
Employee = collections.namedtuple('Employee', ['name', 'id'])
```

The resulting class has extra __annotations__ and _field_types attributes, giving an ordered dict mapping field names to types. __annotations__ should be preferred, while _field_types is kept to maintain pre PEP 526 compatibility. (The field names are in the _fields attribute, which is part of the namedtuple API.) Alternative equivalent keyword syntax is also accepted:

```
Employee = NamedTuple('Employee', name=str, id=int)
```

In Python versions <= 3.5 use:

```
Employee = NamedTuple('Employee', [('name', str), ('id', int)])
```

format :str

index :int

value :str

class idena.types.**Epoch**

currentPeriod :str

currentValidationStart :datetime

epoch :int

nextValidation :datetime

class idena.types.**Flip**

hex :str

privateHex :str

class idena.types.**FlipAnswer**

Bases: typing.NamedTuple

Typed version of namedtuple.

Usage in Python versions >= 3.6:

```
class Employee(NamedTuple):
    name: str
    id: int
```

This is equivalent to:

```
Employee = collections.namedtuple('Employee', ['name', 'id'])
```

The resulting class has extra __annotations__ and _field_types attributes, giving an ordered dict mapping field names to types. __annotations__ should be preferred, while _field_types is kept to maintain pre PEP 526 compatibility. (The field names are in the _fields attribute, which is part of the namedtuple API.) Alternative equivalent keyword syntax is also accepted:

```
Employee = NamedTuple('Employee', name=str, id=int)
```

In Python versions <= 3.5 use:

```
Employee = NamedTuple('Employee', [('name', str), ('id', int)])
```

answer :Answer

grade :Grade

hash :str

wrongWords :List[bool]

class idena.types.**FlipAnswers**

Bases: typing.NamedTuple

Typed version of namedtuple.

Usage in Python versions >= 3.6:

```
class Employee(NamedTuple):
    name: str
    id: int
```

This is equivalent to:

```
Employee = collections.namedtuple('Employee', ['name', 'id'])
```

The resulting class has extra __annotations__ and _field_types attributes, giving an ordered dict mapping field names to types. __annotations__ should be preferred, while _field_types is kept to maintain pre PEP 526 compatibility. (The field names are in the _fields attribute, which is part of the namedtuple API.) Alternative equivalent keyword syntax is also accepted:

```
Employee = NamedTuple('Employee', name=str, id=int)
```

In Python versions <= 3.5 use:

```
Employee = NamedTuple('Employee', [('name', str), ('id', int)])
```

answers :List[FlipAnswer]

class idena.types.**FlipHashes**

available :bool

extra :bool

hash :str

ready :bool

class idena.types.**FlipWords**

id :int

used :bool

words :List[int]

class idena.types.**Grade**

gradeA = 5

gradeB = 4

gradeC = 3

gradeD = 2

gradeNone = 0

gradeReported = 1

class idena.types.**Identity**

address :str

age :int

availableFlips :int

code :bytes

flipKeyWordPairs :Optional[List[FlipWords]]

flips :Optional[List[str]]

generation :int

invitees :Optional[List[TxAddr]]

invites :int

lastValidationFlags :Optional[List[str]]

madeFlips :int

online :bool

penalty :decimal.Decimal

profileHash :str

pubkey :str

requiredFlips :int

stake :decimal.Decimal

state :str

totalQualifiedFlips :int

totalShortFlipPoints :float

class idena.types.**Invite**

hash :str

key :str

receiver :str

class idena.types.**Peer**

Bases: typing.NamedTuple

Typed version of namedtuple.

Usage in Python versions >= 3.6:

```
class Employee(NamedTuple):
    name: str
    id: int
```

This is equivalent to:

```
Employee = collections.namedtuple('Employee', ['name', 'id'])
```

The resulting class has extra __annotations__ and _field_types attributes, giving an ordered dict mapping field names to types. __annotations__ should be preferred, while _field_types is kept to maintain pre PEP 526 compatibility. (The field names are in the _fields attribute, which is part of the namedtuple API.) Alternative equivalent keyword syntax is also accepted:

```
Employee = NamedTuple('Employee', name=str, id=int)
```

In Python versions <= 3.5 use:

```
Employee = NamedTuple('Employee', [('name', str), ('id', int)])
```

addr :str

id :str

class idena.types.**Profile**

info :bytes

nickname :str

class idena.types.**RawFlip**

privateHex :str

publicHex :str

class idena.types.**SendTxArgs**

Bases: *idena.types.BaseTxArgs*

amount :decimal.Decimal

from_ :str

maxFee :decimal.Decimal

payload :str

tips :decimal.Decimal

to :str

type :TxType

useProto :bool

class idena.types.**Stake**

Bases: typing.NamedTuple

Typed version of namedtuple.

Usage in Python versions >= 3.6:

```
class Employee(NamedTuple):
    name: str
    id: int
```

This is equivalent to:

```
Employee = collections.namedtuple('Employee', ['name', 'id'])
```

The resulting class has extra __annotations__ and _field_types attributes, giving an ordered dict mapping field names to types. __annotations__ should be preferred, while _field_types is kept to maintain pre PEP 526 compatibility. (The field names are in the _fields attribute, which is part of the namedtuple API.) Alternative equivalent keyword syntax is also accepted:

```
Employee = NamedTuple('Employee', name=str, id=int)
```

In Python versions <= 3.5 use:

```
Employee = NamedTuple('Employee', [('name', str), ('id', int)])
```

Hash :str

Stake :decimal.Decimal

class idena.types.**State**

name :str

class idena.types.**Sync**

currentBlock :int

genesisBlock :int

highestBlock :int

syncing :bool

wrongTime :bool

class idena.types.**Transaction**

amount :str

blockHash :str

epoch :int

from_ :str

hash :str

maxFee :str

nonce :int

payload :str

timestamp :int

tips :str

to :Optional[str]

type :str

usedFee :str

class idena.types.**TxAddr**

Address :str

TxHash :str

class idena.types.**TxEvent**

data :List[bytes]

eventName :str

class idena.types.**TxReceipt**

contractAddress :str

error :str

events :List[TxEvent]

from_ :str

gasCost :int

gasUsed :int

success :bool

txHash :str

class idena.types.**TxType**

ActivationTx = 1

BurnTx = 12

ChangeGodAddressTx = 11

ChangeProfileTx = 13

EvidenceTx = 8

InviteTx = 2

KillInviteeTx = 10

KillTx = 3

OnlineStatusTx = 9

SendTx = 0

SubmitAnswersHashTx = 5

SubmitFlipTx = 4

SubmitLongAnswersTx = 7

SubmitShortAnswersTx = 6

- Highly inspired by Web3.py and cookiecutter-pypackage.

CHAPTER 2

Installation

Idena.py can be installed using `pip` as follows:

```
$ pip install idena
```

For the development, clone the repository then:

```
$ pip install -e .
```

CHAPTER 3

Using Idena

This library depends on a connection to an Idena node and there are 2 ways to configure them.

3.1 Calling *client.init*

```
>>> from idena import client
>>> client.init('http://localhost:9009/', 'api-key')
```

3.2 Setting environment variables

Set *IDENA_RPC_NODE* and *IDENA_API_KEY* envars:

```
$ export IDENA_RPC_NODE=http://localhost:9009/
$ export IDENA_API_KEY=api-key
```

CHAPTER 4

Getting Blockchain Info

```
>>> client.blockchain.get_last_block()

Block(coinbase='0xbe854231db69ab042073b7ff8309ae3ee265a40f',
   hash='0xa88e6ab305d7ee311ad2de35338cdbf7e664d860709e5a53f0307baeeaa6f968',
   parentHash='0xe324a208892241e0294e5a6334965660375dda7a3ad8d8a42a5f3f2ef2857a22',
   height=2159398,
   timestamp=1606904853,
   root='0xd874709bdd4c6fcd95e2e531cc07a4ce42ab23334dfd12ceb45350535b36664c',
   identityRoot='0x9f3661f19e13d4860f2f2f1610abbbaf86abc8adf1a2781b371189e683745a97',
   ipfsCid=None,
   transactions=None,
   flags=['OfflinePropose'],
   isEmpty=False,
   offlineAddress='0x0df427ad7e1906ab4fcc5fd31118932256f5dc7a')
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

i

idena, 3
idena.apis, 3
idena.apis.account, 3
idena.apis.blockchain, 4
idena.apis.contract, 5
idena.apis.dna, 6
idena.apis.flip, 8
idena.apis.net, 9
idena.client, 9
idena.exceptions, 9
idena.types, 10

Index

Symbols

__repr__() (*idena.exceptions.IdenaException method*), 10

A

activate_invite() (*in module idena.apis.dna*), 7
activate_invite_to_random_address() (*in module idena.apis.dna*), 7
ActivateInviteToRandomAddr (*class in idena.types*), 10
ActivationTx (*idena.types.TxType attribute*), 18
add_peers() (*in module idena.apis.net*), 9
addr (*idena.types.Peer attribute*), 16
address (*idena.types.ActivateInviteToRandomAddr attribute*), 11
address (*idena.types.BurntCoins attribute*), 11
address (*idena.types.Identity attribute*), 14
Address (*idena.types.TxAddr attribute*), 17
AddressTransactions (*class in idena.types*), 11
age (*idena.types.Identity attribute*), 14
amount (*idena.types.BurntCoins attribute*), 12
amount (*idena.types.SendTxArgs attribute*), 16
amount (*idena.types.Transaction attribute*), 17
Answer (*class in idena.types*), 11
answer (*idena.types.FlipAnswer attribute*), 13
answers (*idena.types.FlipAnswers attribute*), 14
available (*idena.types.FlipHashes attribute*), 14
availableFlips (*idena.types.Identity attribute*), 14

B

Balance (*class in idena.types*), 11
balance (*idena.types.Balance attribute*), 11
BaseTxArgs (*class in idena.types*), 11
become_offline() (*in module idena.apis.dna*), 7
become_online() (*in module idena.apis.dna*), 7
Block (*class in idena.types*), 11
blockHash (*idena.types.Transaction attribute*), 17
burn() (*in module idena.apis.dna*), 7
BurntCoins (*class in idena.types*), 11
BurnTx (*idena.types.TxType attribute*), 18

C

call_contract() (*in module idena.apis.contract*), 6
CeremonyIntervals (*class in idena.types*), 12
change_god_address (*in module idena.apis.dna*), 7
change_profile() (*in module idena.apis.dna*), 8
ChangeGodAddressTx (*idena.types.TxType attribute*), 18
ChangeProfile (*class in idena.types*), 12
ChangeProfileTx (*idena.types.TxType attribute*), 18
check_sync() (*in module idena.apis.blockchain*), 4
code (*idena.types.Identity attribute*), 14
coinbase (*idena.types.Block attribute*), 11
contract (*idena.types.ContractTxReceipt attribute*), 12
contractAddress (*idena.types.TxReceipt attribute*), 18
ContractTxReceipt (*class in idena.types*), 12
create_account() (*in module idena.apis.account*), 3
currentBlock (*idena.types.Sync attribute*), 17
currentPeriod (*idena.types.Epoch attribute*), 13
currentValidationStart (*idena.types.Epoch attribute*), 13

D

data (*idena.types.TxEvent attribute*), 17
delete_flip() (*in module idena.apis.flip*), 8
deploy_contract() (*in module idena.apis.contract*), 6
DynamicArg (*class in idena.types*), 12

E

Epoch (*class in idena.types*), 13
epoch (*idena.types.BaseTxArgs attribute*), 11
epoch (*idena.types.Epoch attribute*), 13
epoch (*idena.types.Transaction attribute*), 17
error (*idena.types.ContractTxReceipt attribute*), 12

error (*idena.types.TxReceipt attribute*), 18
estimate_call() (*in module idena.apis.contract*), 6
estimate_deploy() (*in module idena.apis.contract*), 6
estimate_terminate() (*in module idena.apis.contract*), 6
eventName (*idena.types.TxEvent attribute*), 17
events (*idena.types.TxReceipt attribute*), 18
EvidenceTx (*idena.types.TxType attribute*), 18
export_key() (*in module idena.apis.dna*), 8
extra (*idena.types.FlipHashes attribute*), 14

F

flags (*idena.types.Block attribute*), 11
Flip (*class in idena.types*), 13
flip_words() (*in module idena.apis.flip*), 8
FlipAnswer (*class in idena.types*), 13
FlipAnswers (*class in idena.types*), 13
FlipHashes (*class in idena.types*), 14
flipKeyWordPairs (*idena.types.Identity attribute*), 15
FlipLotteryDuration (*idena.types.CeremonyIntervals attribute*), 12
flips (*idena.types.Identity attribute*), 15
FlipSubmit (*in module idena.types*), 10
FlipWords (*class in idena.types*), 14
format (*idena.types.DynamicArg attribute*), 12
from_ (*idena.types.SendTxArgs attribute*), 16
from_ (*idena.types.Transaction attribute*), 17
from_ (*idena.types.TxReceipt attribute*), 18

G

gasCost (*idena.types.ContractTxReceipt attribute*), 12
gasCost (*idena.types.TxReceipt attribute*), 18
gasUsed (*idena.types.ContractTxReceipt attribute*), 12
gasUsed (*idena.types.TxReceipt attribute*), 18
generation (*idena.types.Identity attribute*), 15
genesisBlock (*idena.types.Sync attribute*), 17
get_accounts() (*in module idena.apis.account*), 3
get_address_pending_transactions() (*in module idena.apis.blockchain*), 4
get_address_transactions() (*in module idena.apis.blockchain*), 4
get_balance() (*in module idena.apis.dna*), 8
get_block_by_hash() (*in module idena.apis.blockchain*), 4
get_block_by_height() (*in module idena.apis.blockchain*), 4
get_burnt_coins() (*in module idena.apis.blockchain*), 4
get_ceremony_intervals() (*in module idena.apis.dna*), 8
get_coinbase_address() (*in module idena.apis.dna*), 8
get_current_process() (*in module idena.apis.dna*), 8
get_epoch() (*in module idena.apis.dna*), 8
get_fee_rate() (*in module idena.apis.blockchain*), 4
get_flip() (*in module idena.apis.flip*), 9
get_identities() (*in module idena.apis.dna*), 8
get_identity() (*in module idena.apis.dna*), 8
get_ipfs_addr() (*in module idena.apis.net*), 9
get_last_block() (*in module idena.apis.blockchain*), 4
get_long_flip_hashes() (*in module idena.apis.flip*), 9
get_mempool() (*in module idena.apis.blockchain*), 4
get_peers() (*in module idena.apis.net*), 9
get_profile() (*in module idena.apis.dna*), 8
get_raw_flip() (*in module idena.apis.flip*), 9
get_raw_tx() (*in module idena.apis.blockchain*), 4
get_short_flip_hashes() (*in module idena.apis.flip*), 9
get_stake() (*in module idena.apis.contract*), 6
get_transaction() (*in module idena.apis.blockchain*), 5
get_transaction_receipt() (*in module idena.apis.blockchain*), 5
Grade (*class in idena.types*), 14
grade (*idena.types.FlipAnswer attribute*), 13
gradeA (*idena.types.Grade attribute*), 14
gradeB (*idena.types.Grade attribute*), 14
gradeC (*idena.types.Grade attribute*), 14
gradeD (*idena.types.Grade attribute*), 14
gradeNone (*idena.types.Grade attribute*), 14
gradeReported (*idena.types.Grade attribute*), 14

H

hash (*idena.types.ActivateInviteToRandomAddr attribute*), 11
hash (*idena.types.Block attribute*), 11
hash (*idena.types.ChangeProfile attribute*), 12
hash (*idena.types.FlipAnswer attribute*), 13
hash (*idena.types.FlipHashes attribute*), 14
hash (*idena.types.Invite attribute*), 15
Hash (*idena.types.Stake attribute*), 17
hash (*idena.types.Transaction attribute*), 17
height (*idena.types.Block attribute*), 11
hex (*idena.types.Flip attribute*), 13
highestBlock (*idena.types.Sync attribute*), 17

I

id (*idena.types.FlipWords attribute*), 14
id (*idena.types.Peer attribute*), 16
idena (*module*), 3

idena.apis (*module*), 3
idena.apis.account (*module*), 3
idena.apis.blockchain (*module*), 4
idena.apis.contract (*module*), 5
idena.apis.dna (*module*), 6
idena.apis.flip (*module*), 8
idena.apis.net (*module*), 9
idena.client (*module*), 9
idena.exceptions (*module*), 9
idena.types (*module*), 10
IdenaException, 10
Identity (*class in idena.types*), 14
identityRoot (*idena.types.Block attribute*), 11
index (*idena.types.DynamicArg attribute*), 13
info (*idena.types.Profile attribute*), 16
init() (*in module idena.client*), 9
Invite (*class in idena.types*), 15
invitees (*idena.types.Identity attribute*), 15
invites (*idena.types.Identity attribute*), 15
InviteTx (*idena.types.TxType attribute*), 18
ipfsCid (*idena.types.Block attribute*), 11
isEmpty (*idena.types.Block attribute*), 11
isValidationReady() (*in module idena.apis.dna*), 8

K

key (*idena.types.ActivateInviteToRandomAddr attribute*), 11
key (*idena.types.BurntCoins attribute*), 12
key (*idena.types.Invite attribute*), 15
kill_identity (*in module idena.apis.dna*), 7
kill_invitee (*in module idena.apis.dna*), 7
KillInviteeTx (*idena.types.TxType attribute*), 18
KillTx (*idena.types.TxType attribute*), 18

L

lastValidationFlags (*idena.types.Identity attribute*), 15
left (*idena.types.Answer attribute*), 11
lock_account() (*in module idena.apis.account*), 3
LongSessionDuration (*idena.types.CeremonyIntervals attribute*), 12

M

madeFlips (*idena.types.Identity attribute*), 15
maxFee (*idena.types.SendTxArgs attribute*), 16
maxFee (*idena.types.Transaction attribute*), 17

N

name (*idena.types.State attribute*), 17
nextValidation (*idena.types.Epoch attribute*), 13
nickname (*idena.types.Profile attribute*), 16
nonce (*idena.types.Balance attribute*), 11
nonce (*idena.types.BaseTxArgs attribute*), 11
nonce (*idena.types.Transaction attribute*), 17
none (*idena.types.Answer attribute*), 11

O

offlineAddress (*idena.types.Block attribute*), 11
online (*idena.types.Identity attribute*), 15
OnlineStatusTx (*idena.types.TxType attribute*), 18

P

parentHash (*idena.types.Block attribute*), 11
payload (*idena.types.SendTxArgs attribute*), 16
payload (*idena.types.Transaction attribute*), 17
Peer (*class in idena.types*), 15
penalty (*idena.types.Identity attribute*), 15
privateHex (*idena.types.Flip attribute*), 13
privateHex (*idena.types.RawFlip attribute*), 16
Profile (*class in idena.types*), 16
profileHash (*idena.types.Identity attribute*), 15
pubkey (*idena.types.Identity attribute*), 15
publicHex (*idena.types.RawFlip attribute*), 16

R

RawFlip (*class in idena.types*), 16
read_contract_data() (*in module idena.apis.contract*), 6
read_events() (*in module idena.apis.contract*), 6
readonly_call_contract() (*in module idena.apis.contract*), 6
ready (*idena.types.FlipHashes attribute*), 14
receiver (*idena.types.Invite attribute*), 15
requiredFlips (*idena.types.Identity attribute*), 15
right (*idena.types.Answer attribute*), 11
root (*idena.types.Block attribute*), 11

S

send_dna (*in module idena.apis.dna*), 7
send_invite() (*in module idena.apis.dna*), 8
send_raw_tx() (*in module idena.apis.blockchain*), 5
send_tx() (*in module idena.apis.dna*), 8
SendTx (*idena.types.TxType attribute*), 18
SendTxArgs (*class in idena.types*), 16
ShortSessionDuration (*idena.types.CeremonyIntervals attribute*), 12
Stake (*class in idena.types*), 16
stake (*idena.types.Balance attribute*), 11
stake (*idena.types.Identity attribute*), 15
Stake (*idena.types.Stake attribute*), 17
State (*class in idena.types*), 17
state (*idena.types.Identity attribute*), 15
submit_flip() (*in module idena.apis.flip*), 9
submit_long_answers() (*in module idena.apis.flip*), 9

submit_short_answers() (*in module idena.apis.flip*), 9
SubmitAnswersHashTx (*idena.types.TxType attribute*), 18
SubmitFlipTx (*idena.types.TxType attribute*), 18
SubmitLongAnswersTx (*idena.types.TxType attribute*), 18
SubmitShortAnswersTx (*idena.types.TxType attribute*), 18
subscribe_to_contract_event() (*in module idena.apis.contract*), 6
success (*idena.types.ContractTxReceipt attribute*), 12
success (*idena.types.TxReceipt attribute*), 18
Sync (*class in idena.types*), 17
syncing (*idena.types.Sync attribute*), 17

T

terminate_contract() (*in module idena.apis.contract*), 6
timestamp (*idena.types.Block attribute*), 11
timestamp (*idena.types.Transaction attribute*), 17
tips (*idena.types.SendTxArgs attribute*), 16
tips (*idena.types.Transaction attribute*), 17
to (*idena.types.SendTxArgs attribute*), 16
to (*idena.types.Transaction attribute*), 17
token (*idena.types.AddressTransactions attribute*), 11
totalQualifiedFlips (*idena.types.Identity attribute*), 15
totalShortFlipPoints (*idena.types.Identity attribute*), 15
Transaction (*class in idena.types*), 17
transactions (*idena.types.AddressTransactions attribute*), 11
transactions (*idena.types.Block attribute*), 11
TxAddr (*class in idena.types*), 17
TxEvent (*class in idena.types*), 17
txFee (*idena.types.ContractTxReceipt attribute*), 12
txHash (*idena.types.ChangeProfile attribute*), 12
txHash (*idena.types.ContractTxReceipt attribute*), 12
TxHash (*idena.types.TxAddr attribute*), 17
txHash (*idena.types.TxReceipt attribute*), 18
TxReceipt (*class in idena.types*), 17
TxType (*class in idena.types*), 18
type (*idena.types.SendTxArgs attribute*), 16
type (*idena.types.Transaction attribute*), 17

U

unlock_account() (*in module idena.apis.account*), 3
unsubscribe_from_contract_event() (*in module idena.apis.contract*), 6
used (*idena.types.FlipWords attribute*), 14
usedFee (*idena.types.Transaction attribute*), 17
useProto (*idena.types.SendTxArgs attribute*), 16

V

value (*idena.types.DynamicArg attribute*), 13

W

words (*idena.types.FlipWords attribute*), 14
wrongTime (*idena.types.Sync attribute*), 17
wrongWords (*idena.types.FlipAnswer attribute*), 13