

Week16: 10.24 - 10.28

python-sdk

python-conflux `python-conflux-sdk` `web3.py`

API `3.8 <= python version <= 3.10`

```
python -m venv venv
source ./venv/bin/activate
pip install conflux-web3
```

10.24 python-conflux-sdk: Base32Address

Conflux `Base32` `Base32Address` `isinstance` `str` `==` `__eq__`) [Base32Address](#)

```
>>> from cfx_address import Base32Address
>>> address = Base32Address("0x1ecde7223747601823f7535d7968ba98b4881e09", network_id=1)
' cfxtest: aatp533cg7d0agbd87kz48nj1mpnkca8be1rz695j4'
>>> address_ = Base32Address(address, network_id=1029, verbose=True)
>>> address_
' CFX: TYPE. USER: AATP533CG7D0AGBD87KZ48NJ1MPNKCA8BE7GGP3VPU'
>>> isinstance(address_, str)
True
>>> address_ == ' cfxtest: aatp533cg7d0agbd87kz48nj1mpnkca8be1rz695j4'
True
>>> [
...     address.address_type,
...     address.network_id,
...     address.hex_address,
...     address.verbose_address,
...     address.abbr,
...     address.mapped_evm_space_address,
...     address.eth_checksum_address,
... ]
['user', 1, '0x1ecde7223747601823f7535d7968ba98b4881e09',
```

```
' CFXTEST: TYPE. USER: AATP533CG7D0AGBD87KZ48NJ1MPNKCA8BE1RZ695J4', ' cfxtest: aat...95j4',  
' 0x349f086998cF4a0C5a00b853a0E93239D81A97f6', ' 0x1ECdE7223747601823f7535d7968Ba98b4881E09' ]
```

python-conflux-sdk(conflux-web3) `Base32Address`

`str` / `Base32Address`

```
>>> from conflux_web3 import Web3  
>>> w3 = Web3(Web3.HTTPProvider("https://test.confluxrpc.com"))  
>>> miner = w3.cfx.get_block_by_epoch_number("latest_mined")["miner"]  
>>> miner.address_type  
'user'
```

10.25 python-conflux-sdk: wallet

web3.py `construct_sign_and_send_raw_middleware` `send_transaction`

```
# `web3.py`      `construct_sign_and_send_raw_middleware`  
from web3 import Web3, EthereumTesterProvider  
w3 = Web3(EthereumTesterProvider)  
from web3.middleware import construct_sign_and_send_raw_middleware  
from eth_account import Account  
acct = Account.create('random string')  
w3.middleware_onion.add(construct_sign_and_send_raw_middleware(acct))  
w3.eth.default_account = acct.address  
legacy_transaction = {  
    'to': Account.create('another random string'),  
    'value': 22,  
    'gasPrice': 123456, # optional - if not provided, gas_price_strategy (if exists) or  
    eth_gasPrice is used  
}  
w3.eth.send_transaction(legacy_transaction)
```

python-conflux-sdk `conflux_web3.middleware` `construct_sign_and_send_raw_middleware`)

```
from conflux_web3 import Web3  
from cfx_account import LocalAccount  
w3 = Web3(Web3.HTTPProvider("https://test.confluxrpc.com"))  
acct: LocalAccount = w3.account.create("random string")  
  
w3.cfx.default_account = acct  
# default_account      LocalAccount  
# w3.wallet.add_account(acct)
```

```
# w3.cfx.default_account = acct.address

#           from
#   from       w3.cfx.default_account       from
transaction = {
    'to': w3.address.zero_address(network_id=1),
    'value': 22,
    'gasPrice': 10**9,
}
w3.cfx.send_transaction(transaction).executed()
```

```
# wallet         account
assert acct.address in w3.wallet
acct_ = w3.account.from_key("FILL YOUR SECRET KEY HERE")
w3.wallet.add_account(acct_)
assert (acct.address in w3.wallet) and (acct_ in w3.wallet)

transaction = {
    'to': w3.address.zero_address(network_id=1),
    'value': 22,
    'gasPrice': 10**9,
    'from': acct_.address
}
w3.cfx.send_transaction(transaction).executed()
```

10.26 python-conflux-sdk:

web3.py wait_for_transaction_receipt Conflux

1. pending
2. mined
3. executed 5 epoch
4. confirmed PoW revert
5. finalized PoS revert 5-10

python-conflux-sdk wait_till_transaction_mined, wait_till_transaction_executed, wait_till_transaction_executed, wait_till_transaction_confirmed, wait_till_transaction_finalized

API sdk API

```
from conflux_web3 import Web3
w3 = Web3(Web3.HTTPProvider("https://test.confluxrpc.com"))
acct = w3.account.create("random string")
```

```

w3.cfx.default_account = acct

transaction = {
    'to': w3.address.zero_address(network_id=1),
    'value': 22,
    'gasPrice': 10**9,
}

tx_hash = w3.cfx.send_transaction(transaction)

tx_hash.mined()
tx_hash.executed()
tx_hash.confirmed()
tx_hash.finalized()

```

10.27 python-conflux-sdk: contract name

python-conflux-sdk `name`, `abi`, `bytecode`, `address`

```

from conflux_web3 import Web3

w3 = Web3(Web3.HTTPProvider("https://test.confluxrpc.com"))
acct = w3.account.create()
w3.default_account = acct

# CFX
faucet = w3.cfx.contract(name="Faucet")
assert faucet.address == "cfxtest:acejjfa80vj06j2jgtz9pngkv423fhkuxj786kjr61"
faucet.functions.claimCfx().transact().executed()

# ERC20
erc20_factory = w3.cfx.contract(name="ERC20")
contract_address = (erc20_factory.constructor(name="Coin", symbol="C", initialSupply=10**18)
                    .transact()
                    .executed()["contractCreated"])

# ERC20
erc20_instance = erc20_factory(contract_address)
erc20_instance.functions.transfer(w3.cfx.address.zero_address(network_id=1),
100).transact().executed()

```