



ONTology DAPP开发分享

DAPP开发三要素

私链搭建

1.**Cyano Wallet**(Chrome插件钱包下载地址<https://chrome.google.com/webstore/search/Cyano%20Wallet?hl=zh-CN>)

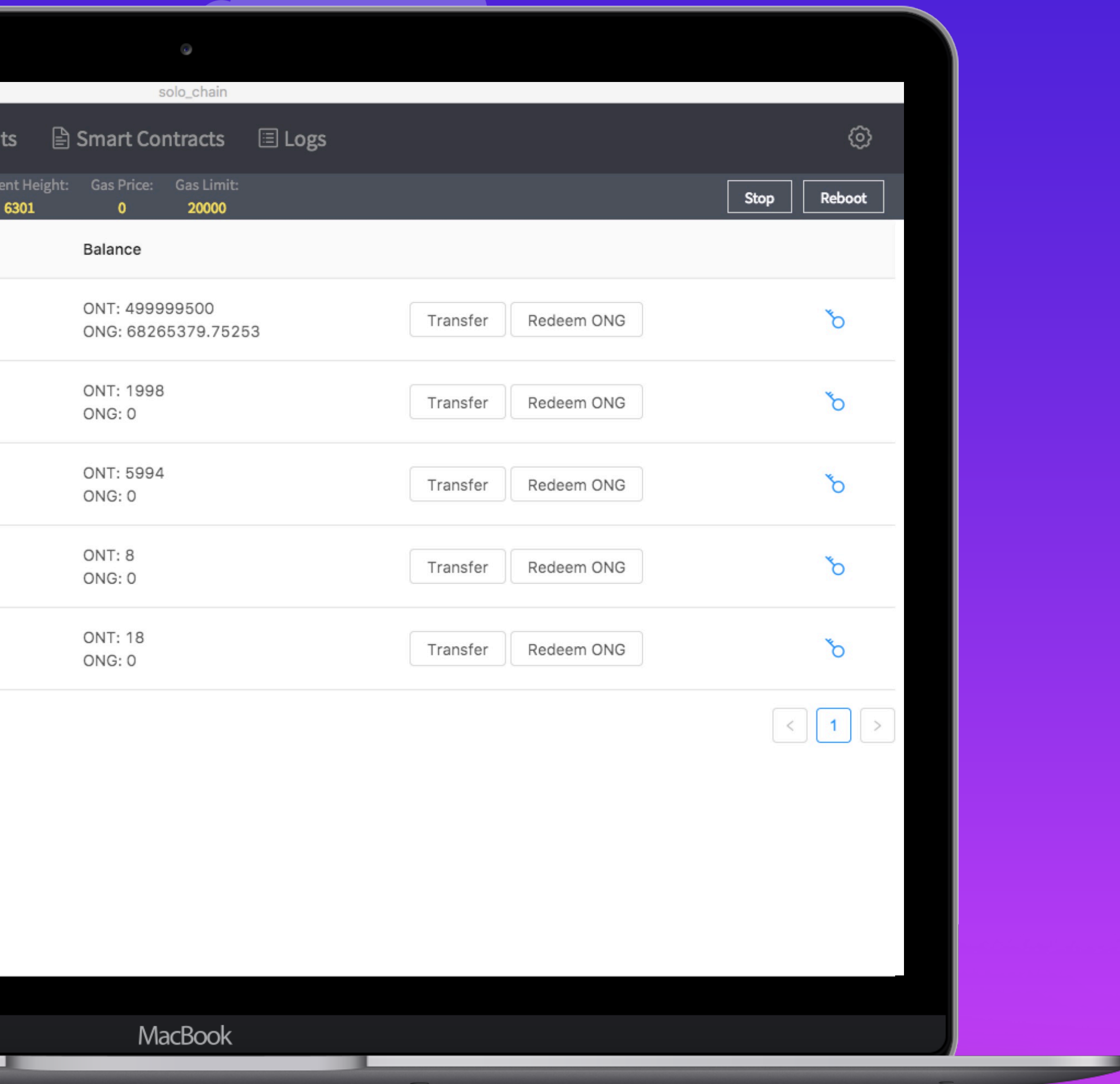
2.**solo-chain**(搭建私链下载地址<https://github.com/punicasuite/solo-chain/releases>)

合约编写

SmartX(智能合约IDE)
地址(<https://smartx.ont.io>)

和前端交互

DAPP和前端交互官方提供文档(<https://github.com/ontio-cyano/integration-docs>)



solo-chain

Cyano Wallet钱包选择测试网,刷新solo-chain则会自动生成5个账户,第一个账户默认有10亿ONT



合约编写核心API

`Storage.CurrentContext` 获取当前储存区上下文

`Storage.Get(StorageContext, Key)` 查询储存区消息

`Storage.Put(StorageContext, Key, Value)` 储存消息

`Storage.Delete(StorageContext, Key)` 删除操作

API地址[https://dev-docs.ont.io/
#/docs-cn/QuickGuide/02-
deploy-invoke](https://dev-docs.ont.io/#/docs-cn/QuickGuide/02-deploy-invoke)

“合约交易相关”

导入哈希字符串 `from boa.interop.Ontology.Runtime import Base58ToAddress`
导入本合约地址 `from boa.interop.System.ExecutionEngine import GetExecutingScriptHash`
引入官方交易合约 `OntContract = Base58ToAddress("AFmseVrdL9f9oyCzZefL9tG6UbvhUMqNMV")`

用户向用户转账

```
param = state(Base58ToAddress(发送地址), Base58ToAddress(接收地址), 交易金额)
res = Invoke(0, OntContract, "transfer", [param])
```

用户向合约转账

```
本合约地址 selfContractAddress = GetExecutingScriptHash()
param = state(Base58ToAddress(发送地址), selfContractAddress, 交易金额)
res = Invoke(0, OntContract, "transfer", [param])
```

合约向用户转账

```
本合约地址 selfContractAddress = GetExecutingScriptHash()
param = state(selfContractAddress, Base58ToAddress(接收地址), 交易金额)
res = Invoke(0, OntContract, "transfer", [param])
```

一个简单的合约代码

包含存取数据和交易

ONT-cryptomeetup合约代码地址(<https://github.com/crypto-meetup-dev/cryptomeetup-ont/blob/master/contract.py>)

```
# 倒入存取相关api
from boa.interop.System.Storage import Put, Get, GetContext
# 倒入返回以及序列化和反序列化api
from boa.interop.System.Runtime import Notify, Serialize, Deserialize
# 导入解析字符串API
from boa.interop.Ontology.Runtime import Base58ToAddress
# 声明转账合约
OntContract = Base58ToAddress("AFmseVrdL9f9oyCzZefL9tG6UbvHUMqNMV")

def Main(operation, args):
    if operation == 'get':
        return get(args[0])
    if operation == 'set':
        return set(args[0], args[1])
    if operation == 'buy':
        return buy(args[0], args[1])

def set(key, value):
    Put(GetContext(), key, Serialize(value))

def get(key):
    valueDeserialize = Get(GetContext(), key)
    value = Deserialize(valueDeserialize)
    Notify([value])

def buy(fromAcc, toAcc):
    param = state(Base58ToAddress(fromAcc), Base58ToAddress(toAcc), 10)
    res = Invoke(0, OntContract, "transfer", [param])
    if res != b'\x01':
        Notify("buy error.")
        return False
    else:
        Notify("buy success.")
        return True
```


DAPP和PC前端交互

引入官方库 `import { client } from 'ontology-dapi'`

执行官方库 `client.registerClient({})`

获取数据

```
result = await client.api.smartContract.invokeRead({  
  contract: '合约哈希',  
  gasLimit: 30000,  
  gasPrice: 500,  
  method: 'get' // 合约里面的函数  
  parmeters: [{type: 'String', value: 'name'}]  
})
```

储存或修改数据

```
result = await client.api.smartContract.invoke({  
  contract: '合约哈希',  
  gasLimit: 30000,  
  gasPrice: 500,  
  method: 'set' // 合约里面的函数  
  parmeters: [{  
    type: 'String', value: 'name'  
    type: 'String', value: 'amz'}]  
})
```



Thank You!

ONT币聚github地址

<https://github.com/crypto-meetup-dev/cryptomeetup-ont>