

本课程文字讲义中涉及到知识点和概念的使用参见视频教程，您可通过以下方式联系并加入[开发者引擎社区](#)获取：



 **Developer Engine**
开发者引擎社区

- Developer Engine 致力于构建和完善区块链行业的生态社区开发与维护，始终坚持以技术为导向，潜心孵化和扶持技术开发者进行底层技术的研究与突破。
- 成立社区初心为了扶持有志向加入web3.0生态建设的开发者加入生态，为开发者提供多元化学习和交流的平台。

 : 开发者引擎
: Metasharing实验室
(干货文章分享)

 : DevEngine
(社区公益开发课程)

 : DevEngine中国
(社区公益开发课程)

 :
@DevEngineChina

➢ 帮助技术开发者由web2转向web3。一群志同道合的小伙伴，一起参与学习、培训、参与生态比赛、发起公链项目，同时还会有后续的职业机会。

➢ 社区不定期录制课程免费提供给大家，帮助大家避坑，少踩坑。

➢ 有编程基础的小伙伴欢迎扫码加微信
➢ 加入社区，学习公链开发



第六课和第七课 CryptoDappy 实战开发

本节课主要分为三个部分：

- 1、项目准备知识
- 2、脚本和交易的交互使用
- 3、CryptoDappy 应用开发演示

一、项目准备知识

1.1、工具网站介绍

以下介绍三个测试工具网站。

- 1、查看测试网合约交易等信息 <https://flow-view-source.com/testnet/>

flow-view-source

Select environment:

- ☒ testnet STATUS
☐ mainnet STATUS
☐ canarynet STATUS

Account

Transaction

Event

如上界面，这个网站不仅可以查询测试网，还可查询主网和 **canarynet** 网站。在这个网站上可以查询账户、交易以及事件的相关信息。下图为上节课注册的 **Blocto** 钱包生成的账户地址的查询结果：

The screenshot displays the Blocto wallet interface. At the top, there's a navigation bar with 'Authenticate' and a user icon. Below it, the account address '0x08fa7b94d3806198' is shown. The account balance is listed as 500.00100000 FLOW and 123.01411583 FUSD, with 0.00% capacity and 3 keys. The 'Keys' section shows three keys with their respective details:

- Key 0:** c3a193e50dd062fd91fcbc4057b3d13ef732da0d7e7ffaa, Weight: 999/1000, Curve: ECDSA_secp256k1, Hash: SHA3_256, Sequence Number: 0.
- Key 1:** b29f8601c3c33096d46ea1eff6a50a6927fa88ea5f8d051, Weight: 1000/1000, Curve: ECDSA_secp256k1, Hash: SHA3_256, Sequence Number: 0.
- Key 2:** 9daab60bf89830043d5e978344406fef5d4c876ce820565, Weight: 1/1000, Curve: ECDSA_secp256k1, Hash: SHA3_256, Sequence Number: 0.

如上账户，有 500FLOW 和 123FUSD，前者为 Flow 链上的代币，后者为 Flow 上发行的稳定币，用美元 1: 1 质押，给已有账户充值用到下个网站。很明显，**Blocto** 钱包生成的账户包含三个密钥对，网站展示的是 3 个公钥，3 个私钥由 **Blocto** 保存着。同时，这三个公钥的权重有所不同，根据前述课程提及的“给交易签名的公钥的权重应大于等于 1000”，所以作者推测 **Blocto** 签名的策略可能包含使用公钥 0 和公钥 2 联合签名。

2、FlowFaucet, Flow 水龙头 <https://testnet-faucet.onflow.org/>

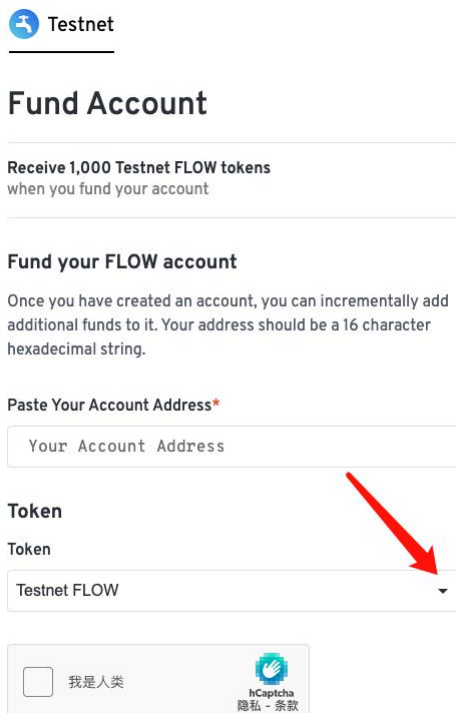
此网站有两个功能：创建测试网账户(CREATE ACCOUNT)，以及给现有的测试网账户充值(FUND ACCOUNT)，两个功能通过网站右上角的按钮切换。

2.1) 新建账户

The screenshot shows the 'Create Account' page on the Flow Faucet website. It includes a header with the 'testnet' tab selected. The main heading is 'Create Account'. Below it, a message states: 'Receive 1,000 Testnet FLOW tokens after your account has been generated!'. The 'Public Key' section explains that the public key must be a 128 character hexadecimal string and provides a link to the Flow CLI. There is a text input field labeled 'Your Public Key'. The 'Signature & Hash Algorithms' section instructs users to select the signature and hash algorithm used to generate their keys, with defaults of ECDSA_P256 and SHA3_256. At the bottom, there are two dropdown menus: 'Signature Algorithm' set to 'ECDSA_P256' and 'Hash Algorithm' set to 'SHA3_256'.

如上图，同样需要先生成密钥对，然后提供公钥，尤其注意签名算法和哈希算法要一致选择。

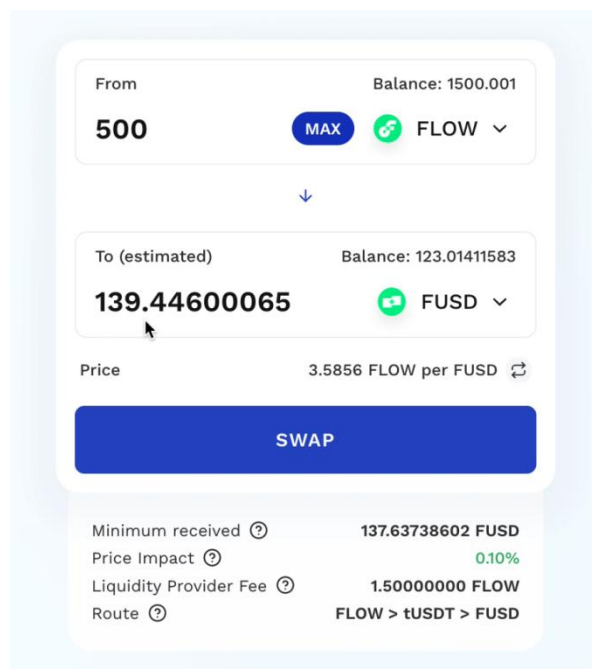
2.2) 充值账户



如上图，只需提供要充值的账户地址，再选择要充值的币种，此网站支持充值 FLOW 和 FUSD 两种币种。建议这里充值 FLOW。然后使用下个网站兑换为 FUSD。

3、FLOW 与 FUSD 兑换 <https://swap-testnet.blocto.app/#/swap>

打开网站后，首先连接钱包，然后进行如下图兑换即可，等待交易完成。



1.2、再谈 CryptoDappy 教程

上一节课已对 CryptoDappy 做了简单的介绍，本节课再对其每个任务详细拆解一下，以下标红的内容为第三部分演示的操作，其他不演示的留作作业或自习。

- 1、Mission #1: 账户登录钱包授权; (上节课已演示)
- 2、Mission #2: **script** 获取 **template** 列表;
- 3、Mission #3: 带参数 **script** 检查 **Collection**, 交易创建和删除 **Collection**;
- 4、Mission #4: 带参数 **script** 获取账户余额, 交易初始化 **Vault**, 介绍测试网 **flow-faucet**;
- 5、Mission #5: 带参数 **script** 获取账户下 **Dappy**, 带参数交易付费 **mint Dappy**, 跟踪监控交易状态变化。
- 6、Mission #6: Dappy-Pack 相关逻辑, 包括 **script** 列表展示, 批量 **mint Dappy**。

1.3、CryptoDappy 项目合约介绍

本项目不需要编写合约, 涉及三个合约均已部署到测试网上, 并已有一些初始化数据。**DappyContract** 为业务主合约, 包含此项目所有的业务逻辑。部署于账户 **0xdb3d539e48a805b7** 下。其他两个为已经部署到测试网上的现有合约, **Fungible Token** 和 **FUSD**, 地址如下:

Source: **FungibleToken.cdc**

NETWORK	CONTRACT ADDRESS
Emulator	0xee82856bf20e2aa6
Testnet	0x9a0766d93b6608b7
Mainnet	0xf233dcee88fe0abe

NETWORK	CONTRACT ADDRESS
Testnet	0xe223d8a629e49c68
Mainnet	0x3c5959b568896393

Source: **FUSD.cdc**

二、脚本和交易的交互使用

这一部分集中介绍一下 **FCL** 中与链上进行交互的主要 **API**, 理论先行, 为第三部分演示做准备。

2.1 脚本(script)交互 query()

这个上节课已经使用过了, 其描述如下。

- 1、入参为单个 **Object**, 最多有如下三个 **key**:

Key	类型	说明
cadence	string, 必填	要执行的有效脚本
args	脚本参数列表	视具体情况而定
limit	数值常量	脚本通常不需要设置

第一个 **cadence** 参数就是 **Cadence** 语言编写的 **Script** 脚本字符串, 用反引号引起来。**args** 是传入脚本的参数; 对于 **query**, **limit** 可以忽略。

- 2、返回值可为任意类型, 也可为空

2.2、交易交互 mutate()

- 1、入参为单个 **Object**, 最多有如下四个 **key**:

前两个 **cadence** 和 **args** 同 **query()** 一样, 分别是 **Cadence** 编写的交易和需要的参数。第三个 **limit**, 对

于交易来说一般都要设置。那 Limit 究竟是什么含义？我们都知道在以太坊上如果要执行一个转账等交易，需要支付一定的手续费，也就是 gas 费。类似的 flow 上的交易也需要，所以有人会理所当然的认为这个 limit 就是指定本次交易最多要付多少 gas 费。如果超过这个值，那干脆就不做这个交易了。真的是这个含义吗？注意不是的，这里的 limit 实际上是告诉 Flow 此次交易计算复杂度的限制，并不是 gas 费的支付上限。至于 Flow 上交易的费用如何计算的，还是有些复杂的，详见官方文档([Variable Transaction Fee](#))，本课程就不再详述了。简单来说，无论是 CyptoDappy 官方的演示，还是其他突进搜索到的交易发起逻辑，基本上设置一个 50~100 的数值，只要你的交易别太复杂，基本上都没什么问题。最后一个就是指定交易的发起人，通常在应用中就是登录钱包的当前账户，这也是这个参数的默认值，可省略。

Key	类型	说明
cadence	string, 必填	要执行的有效交易代码
args	交易参数列表	视具体情况而定
limit	数值常量	一般都要设置
proposer	授权函数	交易发起人，默认是currentUser，一般不需要特殊指定

2、返回值为 交易 ID (transactionId)。

2.3、脚本和交易 args 参数形式

query()和 mutate()函数 args 入参的格式为箭头函数形式，如下，具体的参数类型 FType 的取值参见官方文档。

```
args: (arg, t) => [  
    arg(6, t.Int),  
    arg("name", t.String)  
]
```

2.4、交易状态的追踪 tx()

前已述及，query()用于从链上读取数据，不需要签名验证流程，所以它的执行速度相对来说比较快。而对于 transaction 来说，它的执行就很慢了，要签名，然后还需等待执行，改变链上状态，最后还要出块等等一系列操作，它的肯定是时长要长的。那么从用户角度来看，目前交易到底处到哪一个状态了？交易有没有在运行，还是因为网络出问题交易已经中断了？所以从应用的开发来说，就需要及时的呈现给用户当前交易的状态，让用户清楚交易在正确的进行着。为此，FCL 提供了 tx()，它的入参为 transactionID，即要监控的交易。在 tx()内部会每隔 2500 毫秒采样一次交易的状态。应用开发者可以通过调用如下表的可级联函数获得交易的状态变更的通知返回。

名称	说明
snapshot()	返回当前交易的状态
subscribe(cb)	当状态变更时，调用传入的回调函数cb()
onceFinalized()	当状态为2时返回
onceExecuted()	当状态为3时返回
onceSealed()	当状态为4时返回

而具体的交易状态枚举值如下表所示：

名称	说明
0	未知状态
1	交易发送中
2	发送完毕，等待执行
3	正在执行，等待封印
4	封印完毕
5	交易过期

2.5、通用交互命令 send()和 decode()

send()函数既可以发送脚本也可以发送交易，只需在入参中声明。使用 send()会更加的灵活，如在发送交易时还可指定 authorizer 等参数。另外，send()的返回值必须要首先用 decode()进行解码才可再进一步的解析。通常使用形式如下：

```
fcl.send([...builders]).then(fcl.decode)
```

鉴于 CryptoDappy 官方教程并未使用到 send()，本课不在详细解读，具体使用参考官方文档。

三、CryptoDappy 应用开发演示

本部门演示内容参见视频教程，教程会对关键的前端执行流程进行讲解。

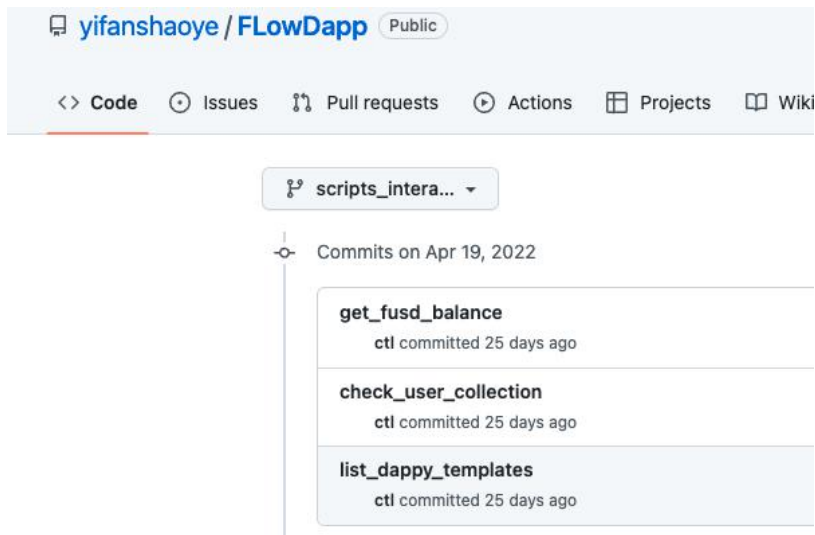
3.1、Script 演示

起始分支：user_wallet_login

成果分支：scripts_interact_display

演示功能：1、无参 script 获取 templates 列表；
2、有参 script 获取账户余额；
3、有参 script 检查账户 collection。

对应代码提交：https://github.com/yifanshaoye/FLOWDapp/commits/scripts_interact_display



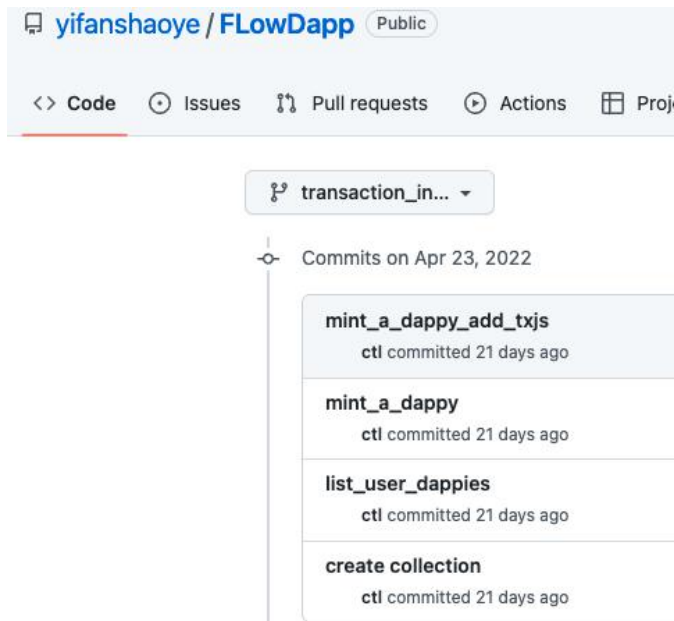
3.2、交易演示

起始分支: scripts_interact_display

成果分支: transaction_interact_display

演示功能: 1、创建 collection;
2、有参 script 获取用户 Dappy;
3、mint 一个 Dappy。

对应代码提交: https://github.com/yifanshaoye/FLowDapp/commits/transaction_interact_display
mint_a_dappy 和 mint_a_dappy_add_txjs 为同一次改动



3.3、监控交易

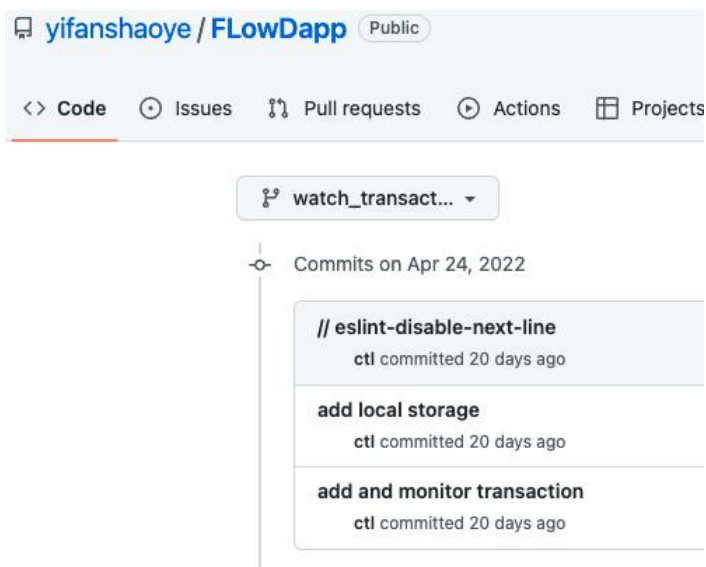
起始分支: transaction_interact_display

成果分支: watch_transaction

演示功能: 1、添加和移除交易;
2、跟踪交易状态变化;
3、实时展示交易状态。

对应代码提交: https://github.com/yifanshaoye/FLowDapp/commits/watch_transaction

最后的// eslint-disable-next-line 提交解决一个前端规范警告的问题



附：CryptoDappy 示例代码各分支成果

分支	功能
main	主分支，最新修改，包含所有功能
create_react_app	初始化 react_app
import_fcl_fetch_bookCount	从链上获取数据(demo 书籍的个数)
cryptodappy_init	cryptodappy 项目初始化，相当于完成前端的开发，准备对接区块链获取数据
user_wallet_login	加入用户登录逻辑
scripts_interact_display	演示 scripts 交互：获取 templates 列表，账户余额，检查 collection
transaction_interact_display	演示交易的交互：创建 Collection，获取用户 Dappy，mint 一个 Dappy
watch_transaction	模拟监控展示交易的状态变化

本节课结束~

至此Flow 生态开发全部课程结束，希望大家有所收获。