

lesson7

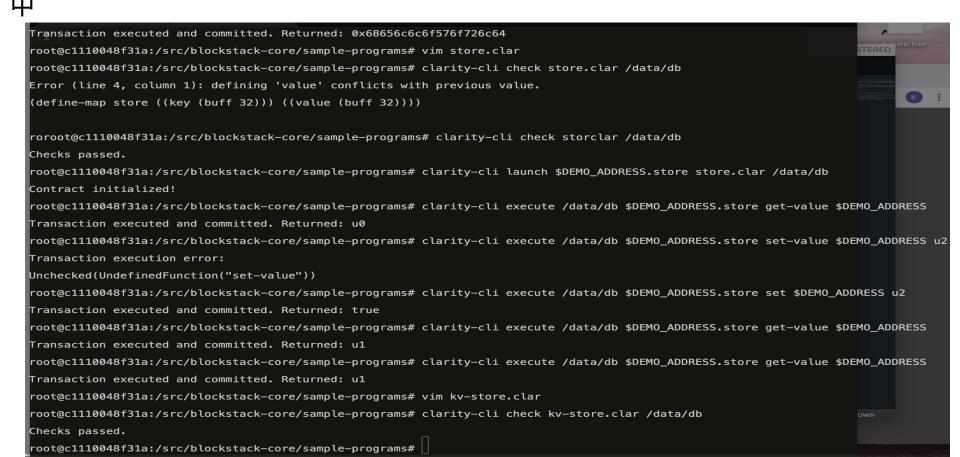
作业

- 搭建Clarity环境, 跑通简单的智能合约并将过程截图提交到screenshot文件夹中
- 分析token.clar代码, 将带有注释的token.clar代码提交到screenshot文件夹中
- 回答思考题, 将思考题答案提交到screenshot文件夹中
- 文件名格式为 "学号-BNS域名"。 例如 "000-gavin.id"

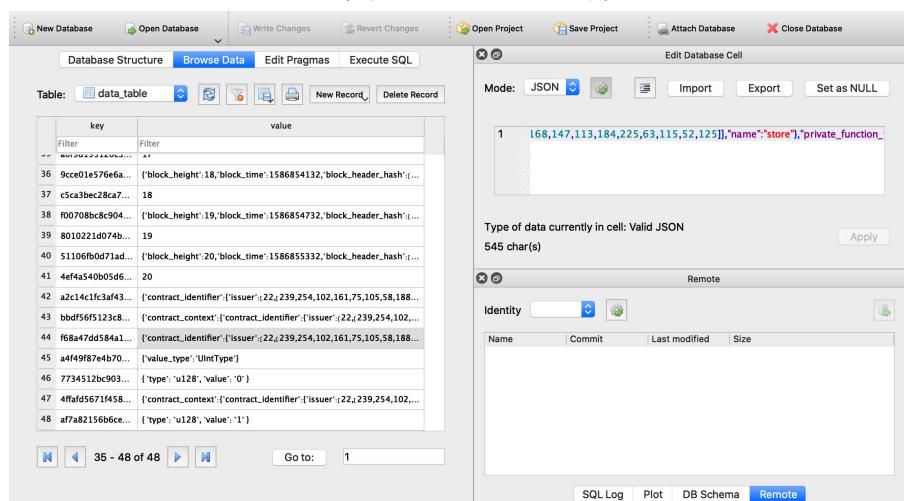
思考题目

- 题目一:根据今天对于智能合约的讲解, 你认为智能合约可以解决哪些现有互联网无法解决的问题?又会带来哪些问题?
- 题目二:前六节课的主要内容均为Blockstack V1的架构, 本节课为Blockstack V2架构中的一个核心内容, 请问你认为V1与V2将如何结合在一起呢?
- 题目三: 如果将本节课的内容应用在去中心化留言板中, 你认为整个留言板的流程图会有什么变化?会添加哪些功能?(建议画出流程图讲解)

1. 搭建Clarity环境,跑通简单的智能合约并将过程截图提交到screenshot文件夹



DB中可以看到已经发布的智能合约资讯



2- 分析token.clar代码,将带有注释的token.clar代码提交到 screenshot文件夹中

```
复杂合约参考
   token.clar
   ; tokens是个mapping, key:account, value: balance
   (define-map tokens ((account principal)) ((balance uint)))
   ; 定义private函数get-balance, 根据account取得此account之balance
    (define-private (get-balance (account principal))
     (default-to u0 (get balance (map-get? tokens (tuple (account account))))))
   ; 定义private函数token-credit!, amount是>0时 principal的balance增加amount值,
   如果 amount <=0 则报错,用在函数token-transfer与mint!中。
   (define-private (token-credit! (account principal) (amount uint))
12
     (if (<= amount u0)
13
         (err "must move positive balance")
         (let ((current-amount (get-balance account)))
14
15
           (begin
             (map-set tokens (tuple (account account))
17
                         (tuple (balance (+ amount current-amount))))
18
             (ok amount)))))
```

```
19
20
    ; 定义public函数token-transfer, tx-sender之balance减少, principal之balance增加
    (define-public (token-transfer (to principal) (amount uint))
21
      (let ((balance (get-balance tx-sender)))
22
        (if (or (> amount balance) (<= amount u0))</pre>
23
24
            (err "must transfer positive balance and possess funds")
25
            (begin
26
              (map-set tokens (tuple (account tx-sender))
                          (tuple (balance (- balance amount))))
27
              (token-credit! to amount))))
28
29
    ; 定义public函数mint! 让tx-sender的balance增加amount值
30
    (define-public (mint! (amount uint))
31
       (let ((balance (get-balance tx-sender)))
32
         (token-credit! tx-sender amount)))
33
```

3. 思考题目

题目一:根据今天对于智能合约的讲解,你认为智能合约可以解决哪些现有互联网无法解决的问题?又会带来哪些问题?

解决的问题: 将中心化的database存储, 改为去中心化的存储, 此外, 可以完全做到流程共有化. 自动化. 不可篡改. 无需中间人服务

带来哪些问题:由于需要给予参与共识机制的所有节点一定的奖励,所以每一笔去中心化的存储都会有一定的成本,此外由于处理共识机制(挖矿)需要时间,TPS预计也会比中心化的存储低一些

题目二: 前六节课的主要内容均为Blockstack V1的架构, 本节课为Blockstack V2架构中的一个核心内容, 请问你认为V1与V2将如何结合在一起呢?

V1主要强调的是DID与user的data存储与隐私与user的定向授权, 群组授权, V2主要强调的是smart contract, 可能可以结合的点会是, 将定向授权与智能合约结合在一起, 例如: 当A付给B某token后, B才将授权码交给A, 可以在生成邀请码的时候通过智能合约setvalue, 然后B想getvalue需要转移token, 目前还需要官方提供javascript端如何与lisp端结合, 最后形成一个完整的dapp。

题目三:如果将本节课的内容应用在去中心化留言板中,你认为整个留言板的流程图会有什么变化?会添加哪些功能?(建议画出流程图讲解)

可以添加若是某user付了token之后,才可以收到某留言群组邀请码的功能。

