

中山大学数据科学与计算机学院本科生实验报告

(2019 年秋季学期)

课程名称：区块链原理与技术

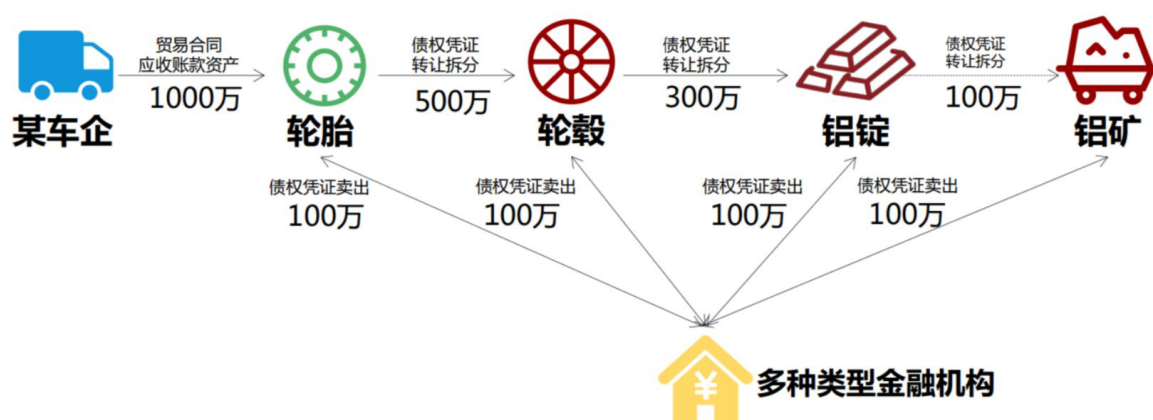
任课教师：郑子彬

年级	17 级	专业（方向）	软件工程
学号	17343057	姓名	雷旭嘉
电话	13246862167	Email	819577710@qq.com
开始日期	2019-11-20	完成日期	2020-1-17

一、项目背景

基于已有的开源区块链系统 FISCO-BCOS (<https://github.com/FISCO-BCOS/FISCO-BCOS>), 以联盟链为主, 开发基于区块链或区块链智能合约的供应链金融平台, 实现供应链应收账款资产的溯源、流转。

场景介绍:



传统供应链金融:

某车企(宝马)因为其造车技术特别牛, 消费者口碑好, 所以其在同行业中占据绝对优势地位。因此, 在金融机构(银行)对该车企的信用评级将很高, 认为他有很大的风险承担的能力。在某次交易中, 该车企从轮胎公司购买了一批轮胎, 但由于资金暂时短缺向轮胎公司签订了 1000 万的应收账款单据, 承诺 1 年后归还轮胎公司 1000 万。这个过程可

以拉上金融机构例如银行来对这笔交易作见证，确认这笔交易的真实性。在接下里的几个月里，轮胎公司因为资金短缺需要融资，这个时候它可以凭借跟某车企签订的应收账款单据向金融机构借款，金融机构认可该车企(核心企业)的还款能力，因此愿意借款给轮胎公司。但是，这样的信任关系并不会往下游传递。在某个交易中，轮胎公司从轮毂公司购买了一批轮毂，但由于租金暂时短缺向轮胎公司签订了 500 万的应收账款单据，承诺 1 年后归还轮胎公司 500 万。当轮毂公司想利用这个应收账款单据向金融机构借款融资的时候，金融机构因为不认可轮胎公司的还款能力，需要对轮胎公司进行详细的信用分析以评估其还款能力同时验证应收账款单据的真实性，才能决定是否借款给轮毂公司。这个过程将增加很多经济成本，而这个问题主要是由于该车企的信用无法在整个供应链中传递以及交易信息不透明化所导致的。

区块链+供应链金融:

将供应链上的每一笔交易和应收账款单据上链，同时引入第三方可信机构来确认这些信息的交易，例如银行，物流公司等，确保交易和单据的真实性。同时，支持应收账款的转让，融资，清算等，让核心企业的信用可以传递到供应链的下游企业，减小中小企业的融资难度。

实现功能:

功能一:实现采购商品—签发应收账款 交易上链。例如车企从轮胎公司购买一批轮胎并签订应收账款单据。

功能二:实现应收账款的转让上链，轮胎公司从轮毂公司购买一笔轮毂，便将于车企的应收账款单据部分转让给轮毂公司。轮毂公司可以利用这个新的单据去融资或者要求车企到期时归还钱款。

功能三:利用应收账款向银行融资上链，供应链上所有可以利用应收账款单据向银行申请融资。

功能四:应收账款支付结算上链，应收账款单据到期时核心企业向下游企业支付相应的欠款。

二、 方案设计

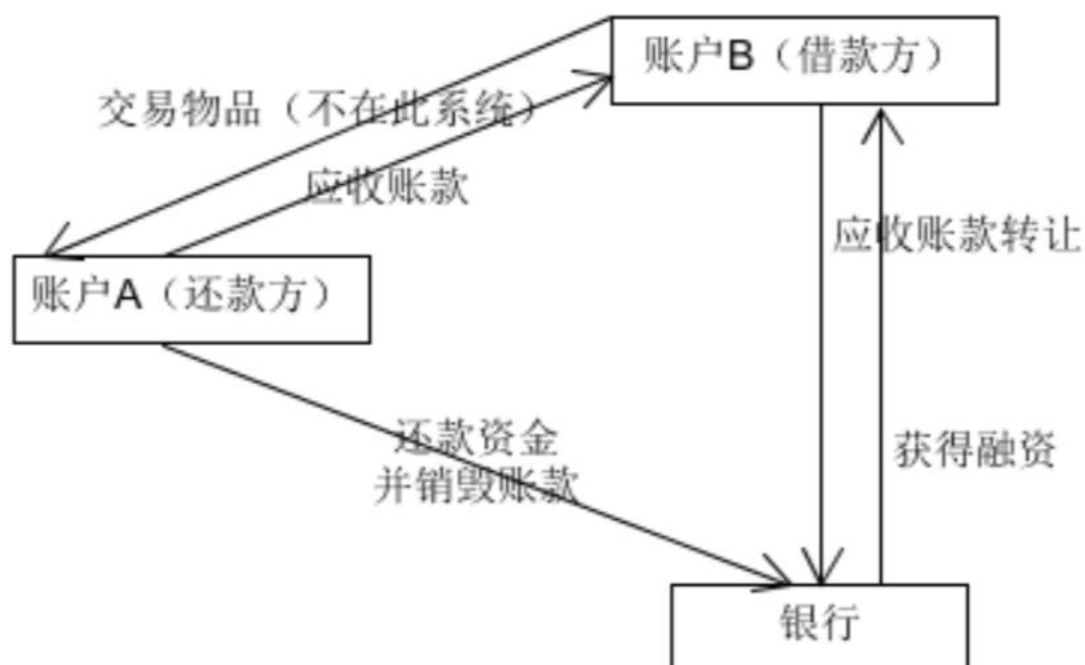
存储设计:

```
4      struct Bill{
5          address borrower;
6          uint amount;
7      }
8      address public bank;
9      mapping (address => uint) private balances;
10     mapping (address => Bill[]) private bill_list;
```

第三方可信机构(银行)的账号地址

每个用户的账号地址和对应的账户余额/应收账款的映射。

数据流图:



所有数据通过合约部署至区块链的形式来储存在区块链上。

核心功能介绍:

功能一:实现采购商品—签发应收账款 交易上链。例如车企从轮胎公司购买一批轮胎并签订应收账款单据。

```
//功能一：签订应收账款单据
function create_bill(address borrower, uint amount) public {
    if (msg.sender == bank) return;
    bill_transfer(borrower, msg.sender, amount);
}
```

通过调用账单转移函数来实现创建新账单功能。两个账户之间单向的多笔交易会合并。

功能二:实现应收账款的转让上链，轮胎公司从轮毂公司购买一笔轮毂，便将于车企的应收账款单据部分转让给轮毂公司。轮毂公司可以利用这个新的单据去融资或者要求车企到期时归还钱款。

```
//功能二：转让单据
function borrow_funds(address borrower, uint amount) public {
    if (msg.sender == bank) return;
    uint i = bill_list[borrower].length;
    do {
        i--;
        if (bill_list[borrower][i].amount > amount){
            bill_list[borrower][i].amount -= amount;
            bill_transfer(bill_list[borrower][i].borrower, msg.sender, amount);
            return;
        }
        else{
            amount -= bill_list[borrower][i].amount;
            bill_transfer(bill_list[borrower][i].borrower, msg.sender, bill_list[borrower][i].amount);
        }
        bill_list[borrower].length--;
        if (amount == 0) return;
    } while (i > 0);
    if (amount != 0)
        create_bill(borrower, amount);
}
```

通过转让还款方手中有的应收账款给借款方，实现债务转移。若应收账款总额不足，则创建新的应收账款以补足差额。

功能三:利用应收账款向银行融资上链，供应链上所有可以利用应收账款单据向银行申请融资。

```
//功能四：结算单据
function remove_bill(address borrower) public {
    for (uint i = 0; i < bill_list[msg.sender].length; i++){
        if (bill_list[msg.sender][i].borrower == borrower){
            balances[borrower] -= bill_list[msg.sender][i].amount;
            if (msg.sender != bank)
                balances[msg.sender] += bill_list[msg.sender][i].amount;
            clear_bill(msg.sender, i);
            return;
        }
    }
}
```

直接把应收账款转让给银行，从而从银行处获得融资。

功能四:应收账款支付结算上链，应收账款单据到期时核心企业向下游企业支付相应的欠款。

```
//功能三：转给银行
function bill_to_bank() public {
    uint amount = 0;
    uint i = bill_list[msg.sender].length;
    do {
        i--;
        amount += bill_list[msg.sender][i].amount;
        bill_transfer(bill_list[msg.sender][i].borrower, bank, bill_list[msg.sender][i].amount);
    } while (i > 0);
    bill_list[msg.sender].length = 0;
    balances[msg.sender] += amount;
}
```

通过资金转移来移除借款人手中的应收账款。

最终成品使用借鉴了 fisco 官网提供的工程项目 <https://github.com/FISCO-BCOS/LargeFiles/raw/master/tools/asset-app.tar.gz>，并在此框架的基础上进行后端的搭建。

三、 功能测试

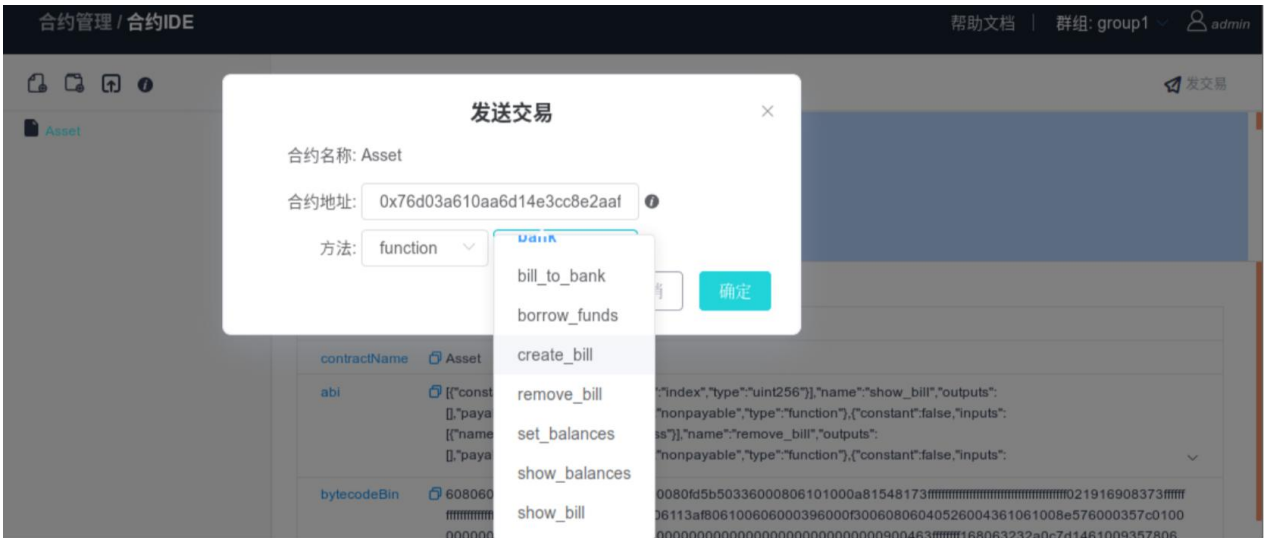
首先，在 webase-web 上的私钥管理中添加四个用户:普通企业用户 A、B、C 分别代表车企，轮胎，轮毂和第三方权威机构 Bank。

用户查看					
新增用户		请输入用户名或公钥地址			
用户名称	用户ID	用户描述	用户公钥地址信息	用户状态	操作
C	700005		0xd5db56696f34e2...	正常	修改
B	700004		0x0c4ecca7648af86...	正常	修改
A	700003		0xd57427fca6543f...	正常	修改
Bank	700002		0x3ab012c3fa543a...	正常	修改

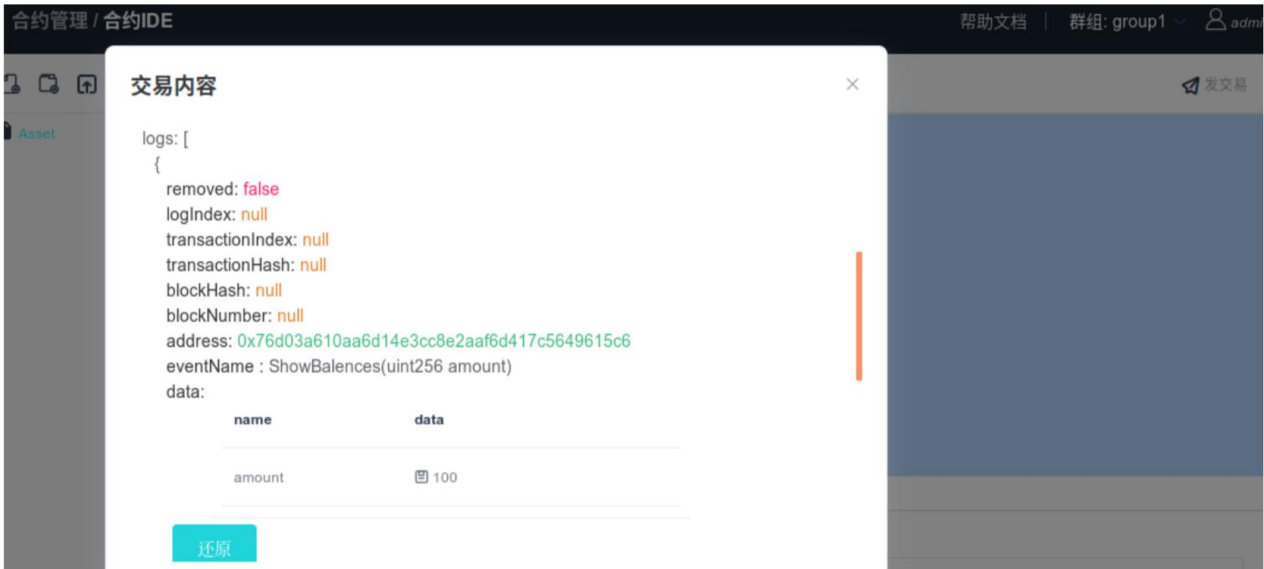
然后上传前文的合约，编译成功后部署到用户 Bank。接下来就可以发送交易验证功能了。

合约管理 / 合约IDE		帮助文档 群组: group1 admin
Asset	<pre>Asset.sol 1 pragma solidity ^0.4.21; 2 contract Asset { 3 4 struct Bill{ 5 address borrower; 6 uint amount; 7 } 8 address public bank; 9 mapping (address => uint) private balances; 10 mapping (address => Bill[]) private bill_list; 11 12 event ShowBalances(uint amount); 13 event ShowBill(address borrower, uint amount); 14 15 constructor() { 16 bank = msg.sender; 17 } 18 19 // 设置账户金额 20 function set_balances(address receiver, uint amount) public {</pre>	发交易

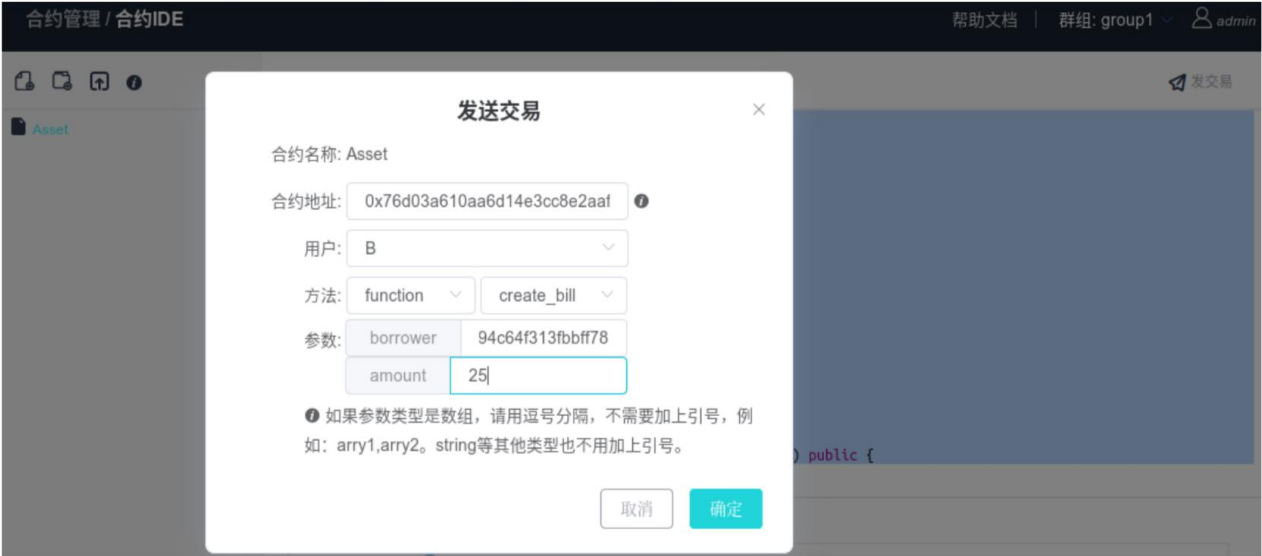
下图中的方法里有 8 个选项;第一个选项为构造函数得到的的 bank 地址;2-5 对应着大作业 s 要求的四个功能;6-8 为辅助验证用的函数，用以设定用户的初始资金和数据显示。



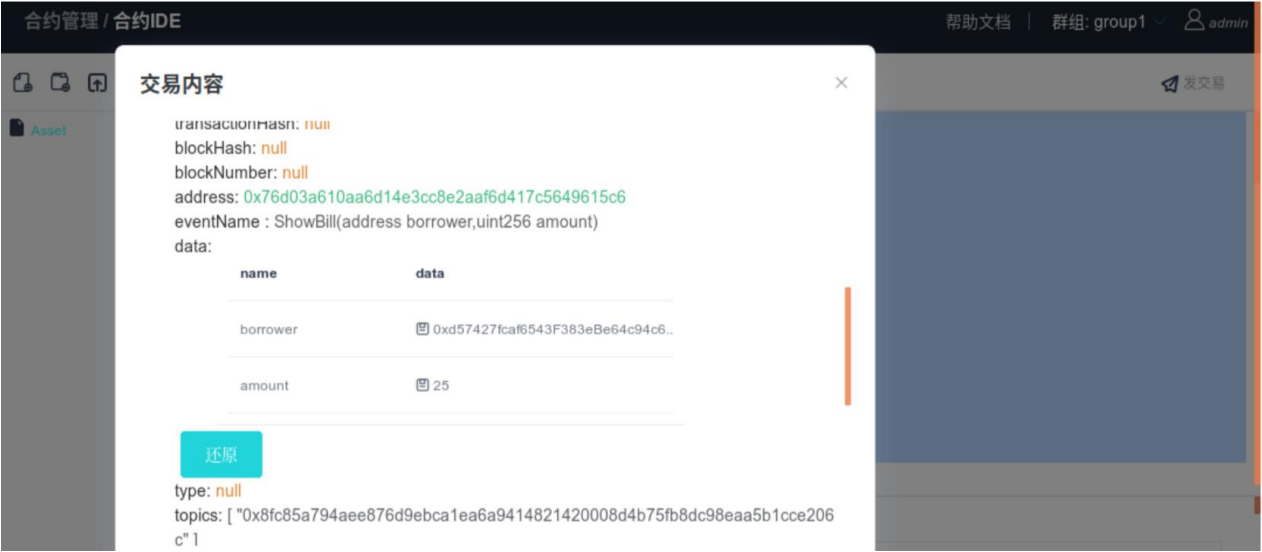
先用 set_balances 设定用户 A、B 的初始资金为 100，C 的初始资金为 0



验证功能一:调用 create_bill 来创建 A 欠下 B 25 资金的应收账款(在实际应用中,即 A/B 的资金无任何变化, A 则通过创建这一应收账款,即可交换到 B 的物品)。



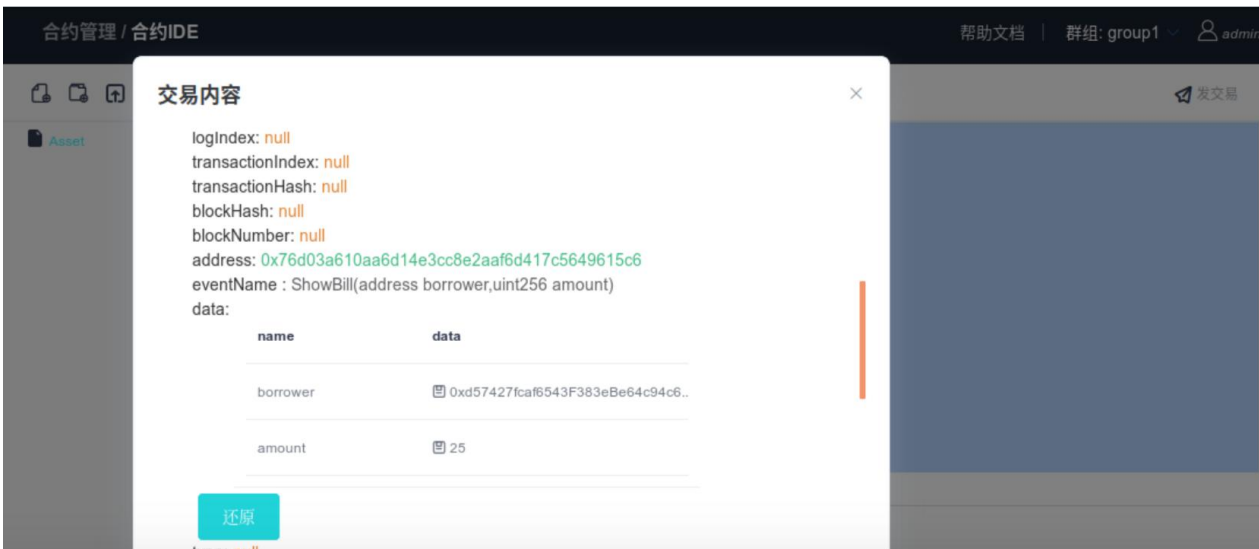
调用 show_bill 来查看 B 的应收账款库, 可以看到 borrower 一栏为 A 的公钥, amount 为应 收金额。功能一验证成功。



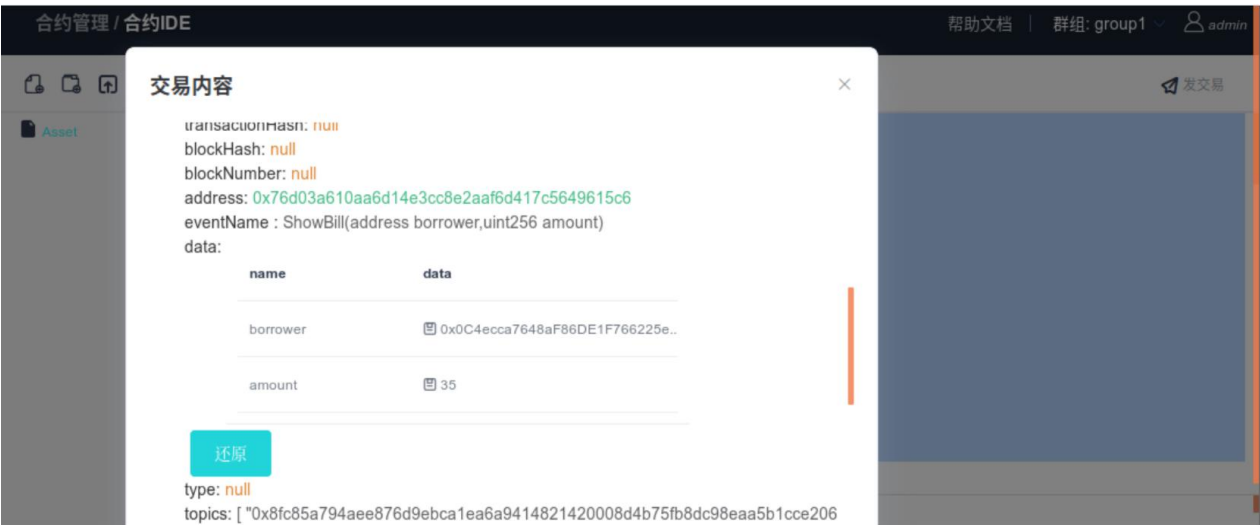
验证功能二:调用 borrow_funds, 让用户 B 向用户 C “借”60 资金(“借”的含义在功能一中有叙述);由于用户 B 在前面已有了一条 25 资金的应收账款, 用户 B 可通过转让这笔应收账款, 从而减少 B 欠下 C 的应收账款。



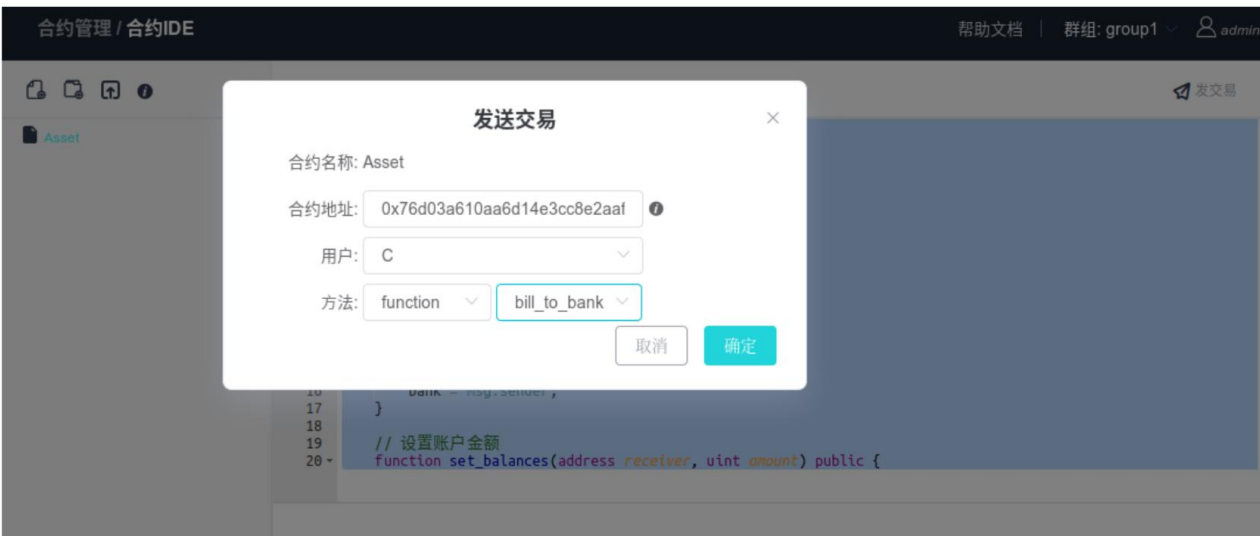
查看 C 的应收账款库, 第一条记录如下所示, 为还款人 A 的应收账款。



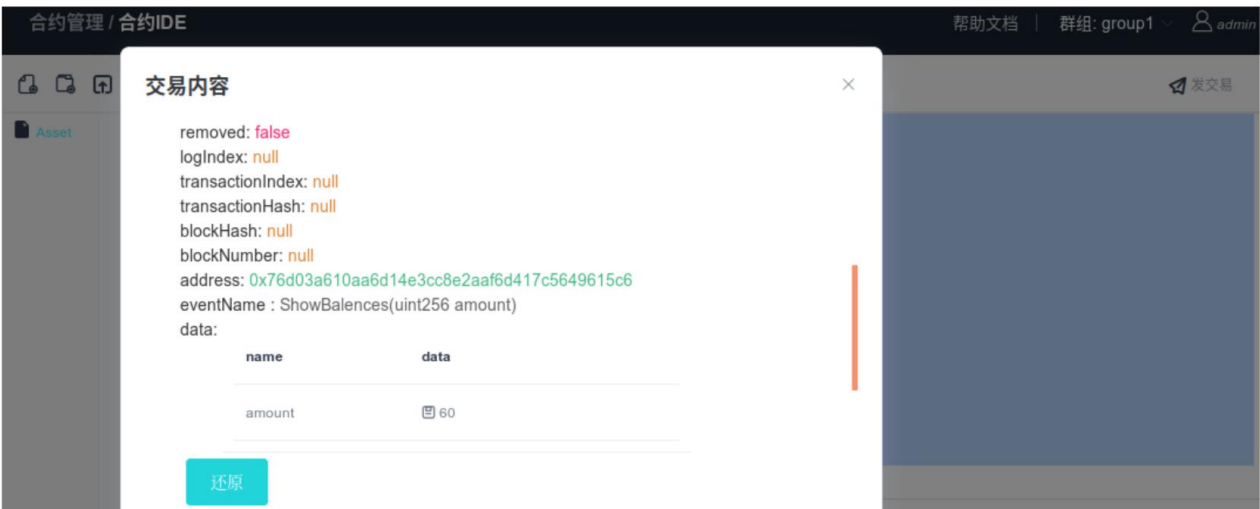
而第二条记录则为还款人 B 的应收账款(转让的应收账款抵消了一部分, 因此 B 应还 $60 - 25 = 35$ 的资金)。功能二验证成功。



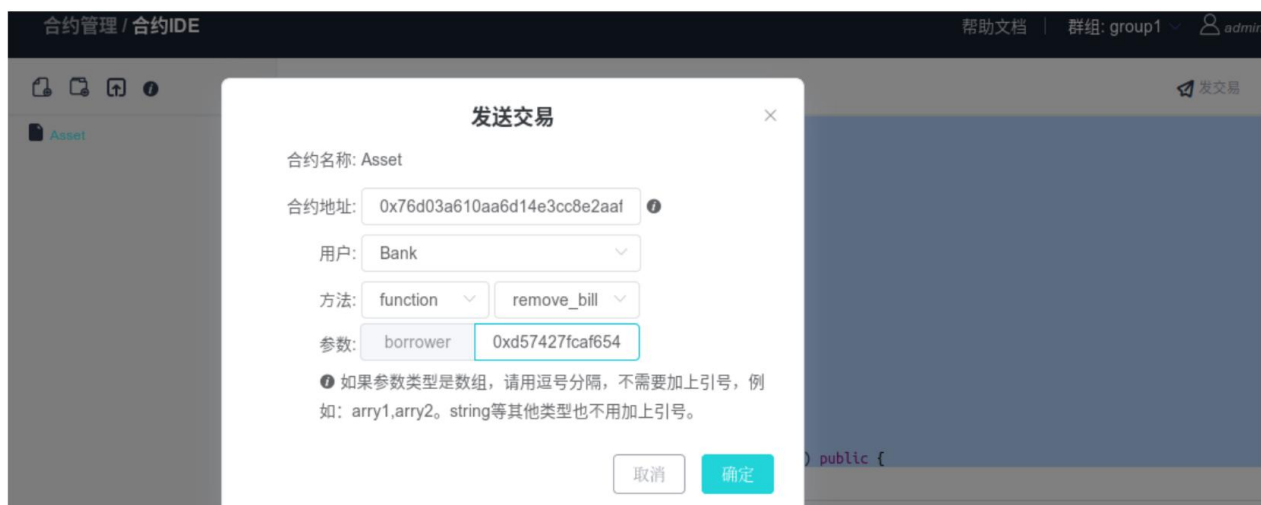
验证功能三 :调用 bill_to_bank, 把 C 的应收账款全部转让给银行来获得融资。



然后查看 C 的资金, 如下图所示, C 获得了 60 资金的融资。功能三验证成功。

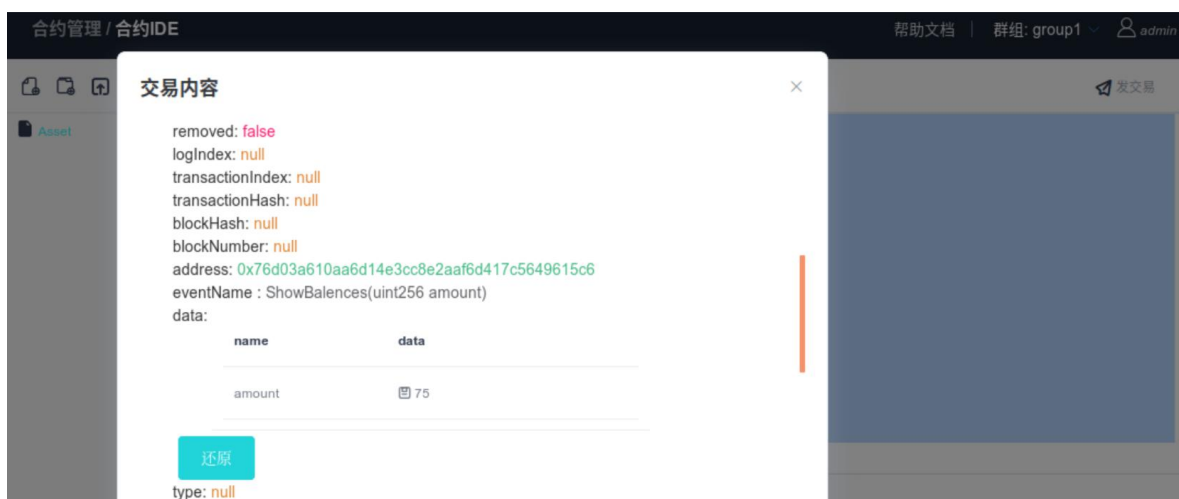


验证功能四 :调用 remove_bill，银行对用户 A 催款。催款成功后移除该应收账款。



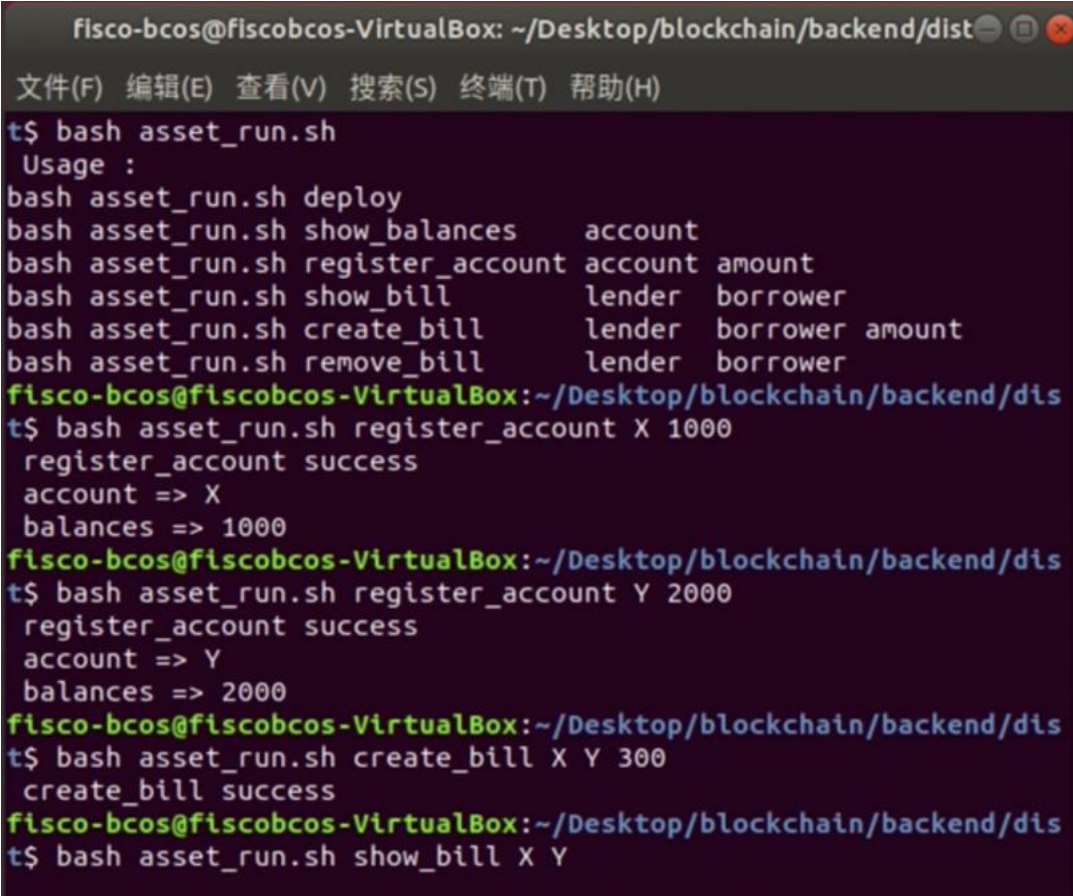
然后查看 A 的资金，如下图，A 的资金为 75，少的 25 资金经由从用户 B 创建的应收账款->用户 B 把该账款转让给用户 C->用户 C 把该账款转让给银行->银行催款结束。

功能四验证成功。



四、 界面展示

受个人能力限制没有实现网页或者其他 API 的展示，此处使用终端代替展示各功能。



```
fisco-bcos@fiscobcos-VirtualBox: ~/Desktop/blockchain/backend/dist
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
t$ bash asset_run.sh
Usage :
bash asset_run.sh deploy
bash asset_run.sh show_balances      account
bash asset_run.sh register_account  account amount
bash asset_run.sh show_bill          lender borrower
bash asset_run.sh create_bill        lender borrower amount
bash asset_run.sh remove_bill        lender borrower
fisco-bcos@fiscobcos-VirtualBox:~/Desktop/blockchain/backend/dis
t$ bash asset_run.sh register_account X 1000
register_account success
account => X
balances => 1000
fisco-bcos@fiscobcos-VirtualBox:~/Desktop/blockchain/backend/dis
t$ bash asset_run.sh register_account Y 2000
register_account success
account => Y
balances => 2000
fisco-bcos@fiscobcos-VirtualBox:~/Desktop/blockchain/backend/dis
t$ bash asset_run.sh create_bill X Y 300
create_bill success
fisco-bcos@fiscobcos-VirtualBox:~/Desktop/blockchain/backend/dis
t$ bash asset_run.sh show_bill X Y
```

五、 心得体会

由于之前没有接触过区块链供应链相关的开发，所以在开发过程中还是经历了许多的挫折与坎坷，也遇到了许多困难，比如未能较好的实现一个优美的前端。在参考了官网提供的工程项目后，逐渐学习尝试了搭建后端，但仍然很大程度增加了我的代码水平。

总而言之通过本次大作业我还是受益匪浅，学到了许多东西。