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	专业（方向）	软件工程
学号	17343080	姓名	陆记
电话	13310237634	Email	1483220884@qq.com
开始日期	2019.11.23	完成日期	2019.12.13

【Github地址】

一、项目背景

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结构借款，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。

整个项目的目的是将核心公司（车企）的还款能力传递下去，使有核心公司的欠款单据的链上的公司可以通过单据向金融机构（银行）借款融资。然而金融机构只认可车企的还款能力，所以当某企业凭单据向金融机构借款时，金融机构需要对该单据的公司进行评估看是否具有还款能力，在本次的情境下，只需要知道单据的源头是否为车企就行了。显然这就是一个简单的溯源问题，只需要追溯单据最初的付款方是否为车企，所以只需要通过区块链的方式将每次交易的信息记录下来就行了。因此，我认为这次需要设计的供应链金融的平台的核心

就是账款单据，我通过一个ability属性来最终表示一个公司是否具有向金融机构借款融资的能力，当然，ability在链上是可以传递的。

二、方案设计

【源码地址】

0. 数据建立

数据结构

公司Company结构体属性：公司名称，资产
单据Bill结构体属性：单据金额，单据交易双方的地址，单据交易时间，单据的还款时间（精确到日），该单据是否被银行认可（ability为True和False），单据的交易次数（transactionCounts）

```
    address car = 0xca35b7d915458ef540ade6068dfe2f44e8fa733c; //汽车公司的地址
    address tyre = 0x14723a09acff6d2a60dcdf7aa4aff308fddc160c; //轮胎公司的地址
    address hub = 0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db; //轮毂公司的地址
    address alui = 0x583031d1113ad414f02576bd6afabfb302140225; //铝锭公司地址
    address alum = 0xdd870fa1b7c4700f2bd7f44238821c26f7392148; //铝矿公司的地址
    address bank = 0x95f452956282a5bfc2c8522bf23cd5525fcbcb9b; //银行地址

    mapping(address => Company) public companyMap; //地址与公司的映射
    mapping(address => Bill) public billMap; // 公司与单据的映射

    struct Bill{
        address payer; //付款人地址
        address payee; //收款人地址
        uint transactionCounts; //交易次数
        uint paymentTime; //支付时间
        uint repaymentTime; //还款时间
        uint billBalance; //余额
        bool ability; //是否具有还款能力
    }

    struct Company{
        bytes32 companyName; //公司名
        uint property; //公司资产
    }
```

event

```
    // 创建单据事件
    event Bill_(string msg, address payer, address payee, uint amount, bool
ability);
    // 单据转让事件
    event Tran_(string msg, address transferer, address receiver, uint amount,
uint payerBillBalance, uint payeeBillBalance, bool payerBillAbility, bool
payeeBillAbility);
```

```
    // 融资事件
    event Fina_(string msg, address company, address financier, uint amount, uint
companyPro);
    // 还款事件
    event Pay_(string msg, address payer, address payee, uint amount, uint
payerPro, uint payeePro);
```

构造函数

- Company创建并初始化

```
    Company public com__ = Company({companyName : "", property : 3000});
    constructor() public{
        // 建立公司
        Company memory car_com = com__;
        car_com.companyName = "Car";
        car_com.property = 10000;
        Company memory tyre_com = com__;
        tyre_com.companyName = "Tyre";
        Company memory hub_com = com__;
        hub_com.companyName = "Hub";
        Company memory alui_com = com__;
        alui_com.companyName = "Alui";
        Company memory alum_com = com__;
        alum_com.companyName = "Alum";
        Company memory bank_com = com__;
        bank_com.companyName = "Bank";
        bank_com.property = 100000;
    }
```

- 映射表的建立

```
        // 建立公司地址与公司的映射
        companyMap[car] = car_com;
        companyMap[tyre]= tyre_com;
        companyMap[hub] = hub_com;

        // 建立公司和bill的映射
        billMap[car] = bill__;
        billMap[tyre] = bill__;
        billMap[hub] = bill__;
```

公司名	地址
Car	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
Tyre	0x14723a09acff6d2a60dcdf7aa4aff308fddc160c

公司名	地址
Hub	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
Alui	0x583031d1113ad414f02576bd6afabfb302140225
Alum	0xdd870fa1b7c4700f2bd7f44238821c26f7392148
Bank	0x95f452956282a5bfc2c8522bf23cd5525fcbcb9b

下面为四个基本功能的实现

1. 交易单据生成并上链

单据生成函数：function createBill(address payee, uint amount) public {}

```
    function createBill(address payee, uint amount) public {
        require(msg.sender != payee, "Bill payee cannot be yourself!");
        require(payee != bank, "Bill payee cannot be bank!");
        Bill memory bill;
        bill.transactionCounts = 1;
        bill.payer = msg.sender;
        if (bill.payer != car) {
            bill.ability = false;
        }
        else bill.ability = true;
        bill.payee = payee;
        bill.billBalance = amount;
        // 创建公司的Bill
        billMap[msg.sender] = bill;
        billMap[payee] = bill;
        emit Bill_("Create a bill!", bill.payer, payee, amount, bill.ability);
    }
```

要求：

- 收款公司不能为单据生成公司
- 收款公司不能为银行

功能：

- 设置交易双方和单据金额（根据传入参数）
- 交易次数（`transactionCounts`）变为1
- 设置交易时间
- 根据payer（付款公司）确认该单据是否有`ability`
- 根据payee（收款公司）确认该单据交易的商品（`commodity`）信息
- 将Bill加入Commpany与Bill的映射表中

2. 单据的转让

函数：function transferBill(address payee, uint amount) public{}

```
    // 转让单据
    function transferBill(address receiver, uint amount) public{
        require(billMap[msg.sender].billBalance >= amount, "Insufficient bill
balance!");
        billMap[msg.sender].payer = msg.sender;
        billMap[msg.sender].payee = receiver;
        billMap[msg.sender].billBalance -= amount;
        billMap[msg.sender].transactionCounts += 1;
        billMap[receiver].transactionCounts += 1;
        billMap[receiver].payer = msg.sender;
        billMap[receiver].payee = receiver;
        billMap[receiver].billBalance += amount;
        billMap[receiver].ability = billMap[msg.sender].ability;
        emit Tran_("Transfer a Bill!", msg.sender, receiver, amount,
billMap[msg.sender].billBalance, billMap[receiver].billBalance,
billMap[msg.sender].ability, billMap[receiver].ability);

    }
```

要求：

- 单据中的交易次数至少为1
- 单据的金额不小于交易金额

功能：

- 创建新的Bill
- 更新交易双方信息
- 付款方单据金额减少
- 收款方单据金额增加
- 交易时间的修改
- 交易次数加1
- `ability`不变
- 将更新的Bill加入到billMap中
  说明：单据的转让过程中，其实交易双方的信息修改和交易商品的更新是不重要的，重要的是`ability`的传递、单据金额的变化和交易时间的更新，`ability`和金额确定公司能否根据此单据向银行借一定资金

3. 凭单据向银行借款融资

函数：function borrowBill(address payee, uint amount) public{}

```
    // 向银行申请融资
    function applyFinancing(address financier, uint amount) public{
        require(financier == bank, "Please apply to the bank for financing");
        require(billMap[msg.sender].ability == true, "Bills are not eligible for
financing!");
        require(billMap[msg.sender].billBalance >= amount, "The amount of
financing exceeds the balance of the bill!");
```

```
        companyMap[msg.sender].property += amount;
        billMap[msg.sender].payer = msg.sender;
        billMap[msg.sender].payee = bank;
        billMap[msg.sender].transactionCounts += 1;
        billMap[msg.sender].billBalance -= amount;
        emit Fina_("Apply to the bank for financing!", msg.sender, financier,
amount, companyMap[msg.sender].property);
    }
```

要求：

- Bill的ability为true
- Bill的金额不小于借款金额
- Bill的收款方必须为银行

功能：

- 单据转让功能
- 公司资金增加
- 金融机构资金减少
- 将更新的Bill加入到billMap中

4. 单据的支付结算

函数：function payBill() public {}

```
    // 还款
    function payBill(address payee) public {
        require(msg.sender == car, "You don't have to pay bill!");
        require(billMap[payee].ability == true && billMap[payee].billBalance > 0,
"You don't have to pay no ability bill!");
        require(companyMap[msg.sender].property > billMap[payee].billBalance,
"Insufficient company assets!");
        uint amount = billMap[payee].billBalance;
        billMap[payee] = bill__;
        companyMap[msg.sender].property -= amount;
        companyMap[payee].property += amount;
        emit Pay_("Pay a bill!", msg.sender, payee, amount,
companyMap[msg.sender].property, companyMap[payee].property);

    }
```

要求：

- 结算公司为Car
- 还款单据的ability为true
- 结算公司的财产（property）大于单据的金额（billBalance）

功能：

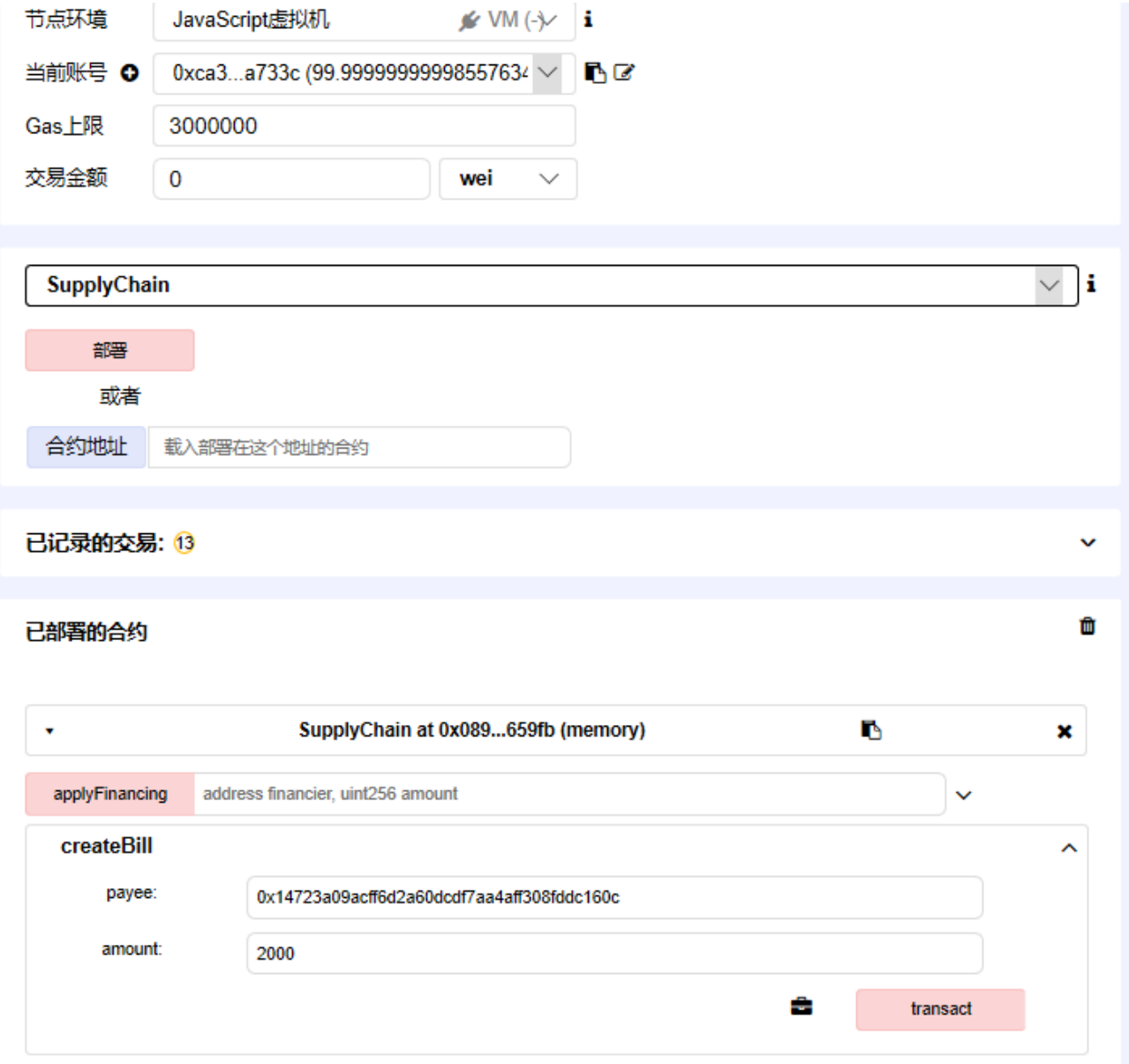
- 单据销毁（重新初始化）
- 持有单据的公司资金增加
- 核心公司的资金减少

三、链端功能测试（Remix）

1、单据生成

- `Car`公司生成`ability`为`true`的单据(能向银行借款)

  调用createBill函数，当前账户为payer（Car），payee为传入的公司地址(Tyre)，amount为单据金额



  函数调用结果

logs	[{ "from": "0x08970fed061e7747cd9a38d680a601510cb659fb", "topic": "0xf78b7abe1ac38ace67b80c2b38cf419976c06b0e63077f33290e2f7cb696fa5a", "event": "Bill_", "args": { "0": "Create a bill!", "1": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c", "2": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "3": "2000", "4": true, "msg": "Create a bill!", "payer": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c", "payee": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "amount": "2000", "ability": true, "length": 5 } }]

查询**"billMap"**中的**Bill**结果**(decoded input** 为**key**：公司地址；**decoded output**为**value**：**Bill**信息**)**

[call] from:0xca35b7d915458ef540ade6068dfe2f44e8fa733c to:SupplyChain.billMap(address) data:0x91b...a733c

transaction hash	0x0e9291f00d0ac228d3b50db9501d57c7f7f8dd6ad136858815d828646208a197
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	SupplyChain.billMap(address) 0x08970fed061e7747cd9a38d680a601510cb659fb
transaction cost	24806 gas (Cost only applies when called by a contract)
execution cost	2126 gas (Cost only applies when called by a contract)
hash	0x0e9291f00d0ac228d3b50db9501d57c7f7f8dd6ad136858815d828646208a197
input	0x91b...a733c
decoded input	{ "address ": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c" }
decoded output	{ "0": "address: payer 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c", "1": "address: payee 0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "2": "uint256: transactionCounts 1", "3": "uint256: paymentTime 0", "4": "uint256: repaymentTime 0", "5": "uint256: billBalance 2000", "6": "bool: ability true" }

- Tyre公司生成ability为false的单据

  调用**createBill**函数，当前账户为**payer**（**Tyre**），**payee**为传入的公司地址**(Car)**，**amount**为单据金额


节点环境
JavaScript虚拟机
VM (-)
当前账号
0x147...c160c (99.9999999999990692
Gas上限
3000000
交易金额
0
wei
SupplyChain
部署
或者
合约地址
载入部署在这个地址的合约
已记录的交易:
14
已部署的合约
SupplyChain at 0x089...659fb (memory)
applyFinancing
address financier, uint256 amount
createBill
payee:
0xca35b7d915458ef540ade6068dfe2f44e8fa733c
amount:
2000
transact

函数调用结果

decoded input	{
	"address payee": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
	"uint256 amount": "2000"
	}

decoded output	{}

logs	[
	{
	"from": "0x08970fed061e7747cd9a38d680a601510cb659fb",
	"topic": "0xf78b7abe1ac38ace67b80c2b38cf419976c06b0e63077f33290e2f7cb696fa5a",
	"event": "Bill_",
	"args": {
	"0": "Create a bill!",
	"1": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C",
	"2": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
	"3": "2000",
	"4": false,
	"msg": "Create a bill!",
	"payer": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C",
	"payee": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
	"amount": "2000",
	"ability": false,
	"length": 5
	}
	}
	]

查询**"billMap"**中的**Bill**结果**(decoded input** 为**key**：公司地址；**decoded output**为**value**：**Bill**信息**)**

[call] from:0x14723a09acff6d2a60dcdf7aa4aff308fddc160c to:SupplyChain.billMap(address) data:0x91b...c160c

transaction hash	0xe7ab1545b0185d4896e52103f125bf2b6b430e2048b46bbb8fd32e897522109e
from	0x14723a09acff6d2a60dcdf7aa4aff308fddc160c
to	SupplyChain.billMap(address) 0x08970fed061e7747cd9a38d680a601510cb659fb
transaction cost	24806 gas (Cost only applies when called by a contract)
execution cost	2126 gas (Cost only applies when called by a contract)
hash	0xe7ab1545b0185d4896e52103f125bf2b6b430e2048b46bbb8fd32e897522109e
input	0x91b...c160c

decoded input	{
	"address ": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C"
	}

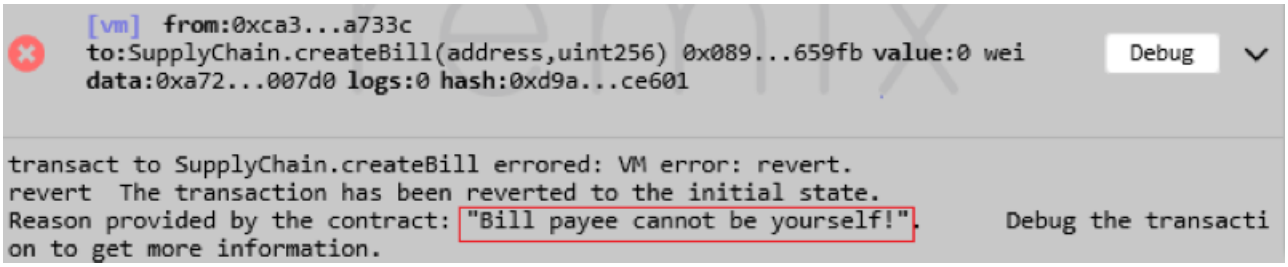
decoded output	{
	"0": "address: payer 0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C",
	"1": "address: payee 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
	"2": "uint256: transactionCounts 1",
	"3": "uint256: paymentTime 0",
	"4": "uint256: repaymentTime 0",
	"5": "uint256: billBalance 2000",
	"6": "bool: ability false"
	}

logs	[]

- 不能生成payee为自身的单据

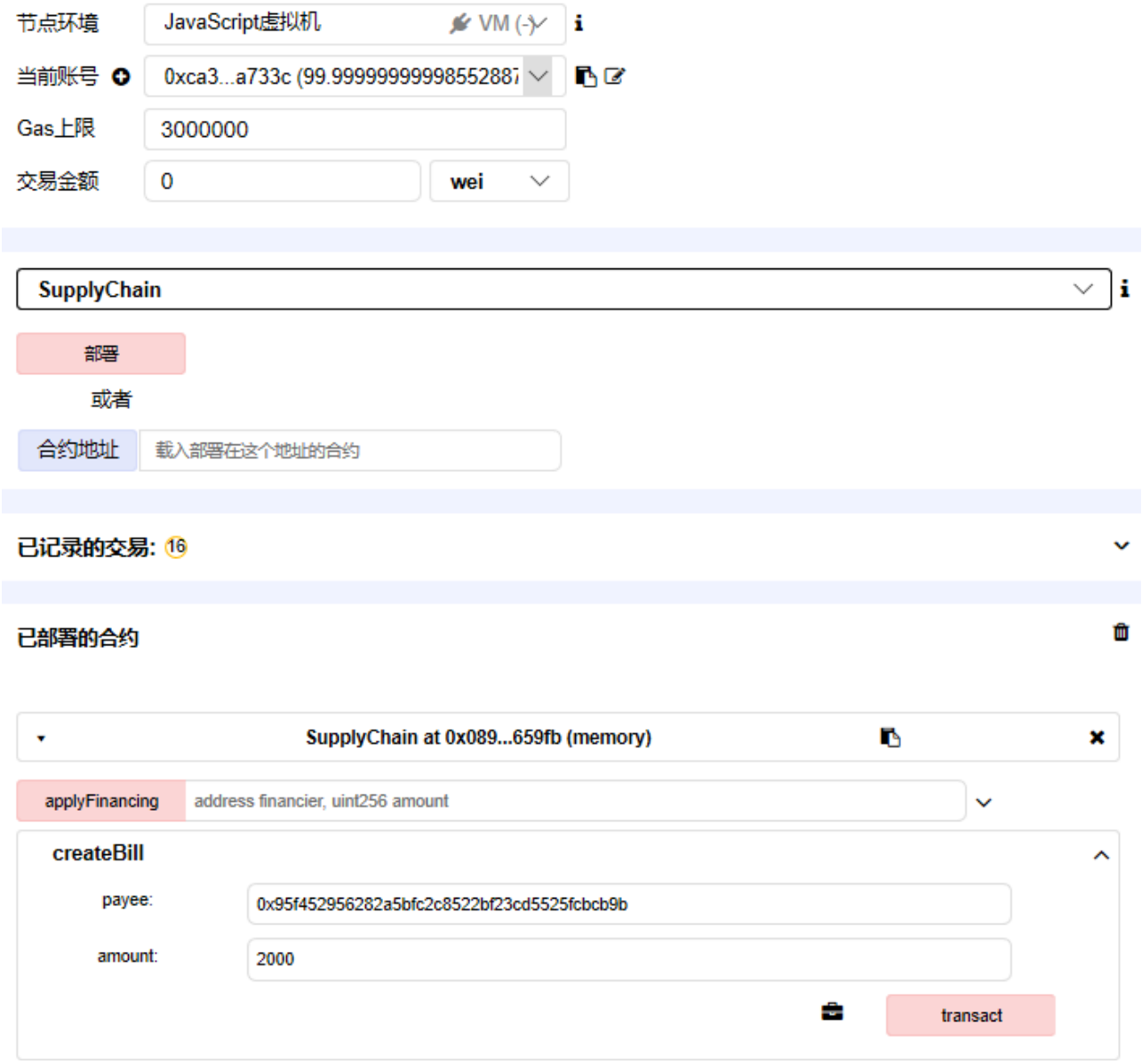
  调用**createBill**函数，当前账户为**payer**（**Car**），**payee**为传入的公司地址，**amount**为单据金额

函数调用结果

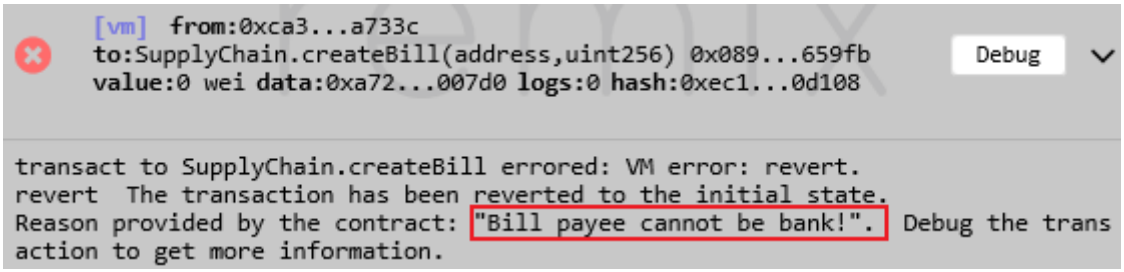


- 不能生成收款公司为Bank的单据

  调用**createBill**函数，当前账户为**payer**（**Car**），**payee**为传入的公司地址**(Bank)**，**amount**为单据金额

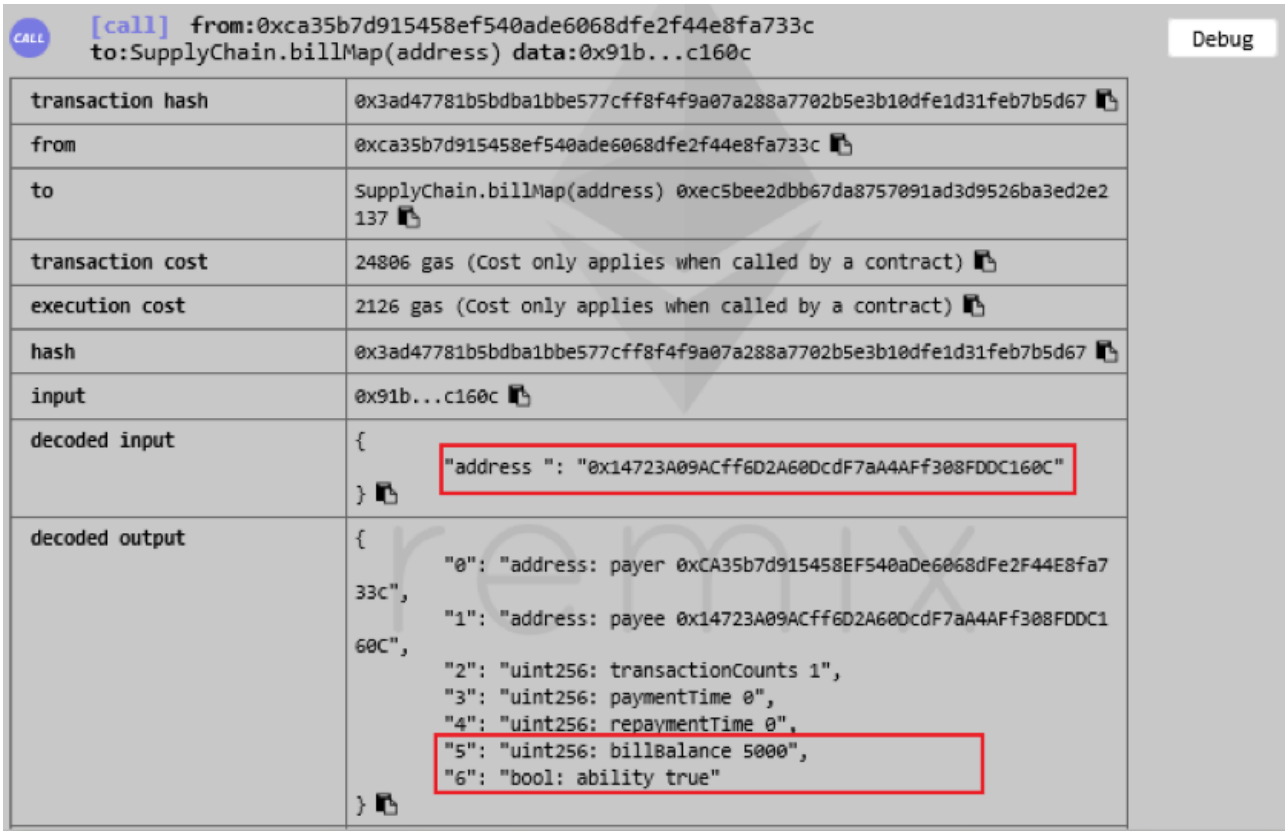


函数调用结果



2、单据转让

- 先调用`billMap`查看原来的`Tyre`公司持有的单据情况（特别是`billBalance`和`ability`）

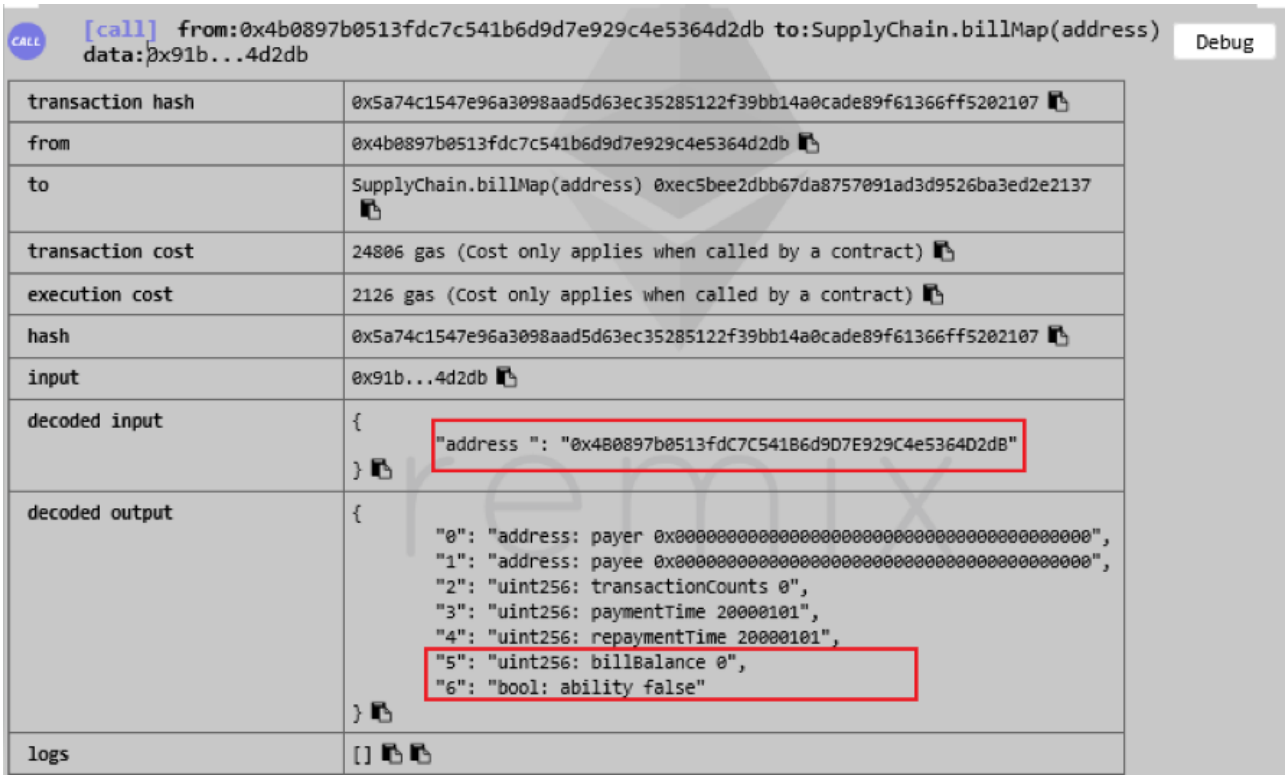


[call] from:0xca35b7d915458ef540ade6068dfe2f44e8fa733c to:SupplyChain.billMap(address) data:0x91b...c160c

Debug

transaction hash	0x3ad47781b5bdba1bbe577cff8f4f9a07a288a7702b5e3b10dfe1d31feb7b5d67
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	SupplyChain.billMap(address) 0xec5bee2dbb67da8757091ad3d9526ba3ed2e2137
transaction cost	24806 gas (Cost only applies when called by a contract)
execution cost	2126 gas (Cost only applies when called by a contract)
hash	0x3ad47781b5bdba1bbe577cff8f4f9a07a288a7702b5e3b10dfe1d31feb7b5d67
input	0x91b...c160c
decoded input	{ "address ": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C" }
decoded output	{ "0": "address: payer 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733C", "1": "address: payee 0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "2": "uint256: transactionCounts 1", "3": "uint256: paymentTime 0", "4": "uint256: repaymentTime 0", "5": "uint256: billBalance 5000", "6": "bool: ability true" }

- 再调用billMap查看Hub公司的单据



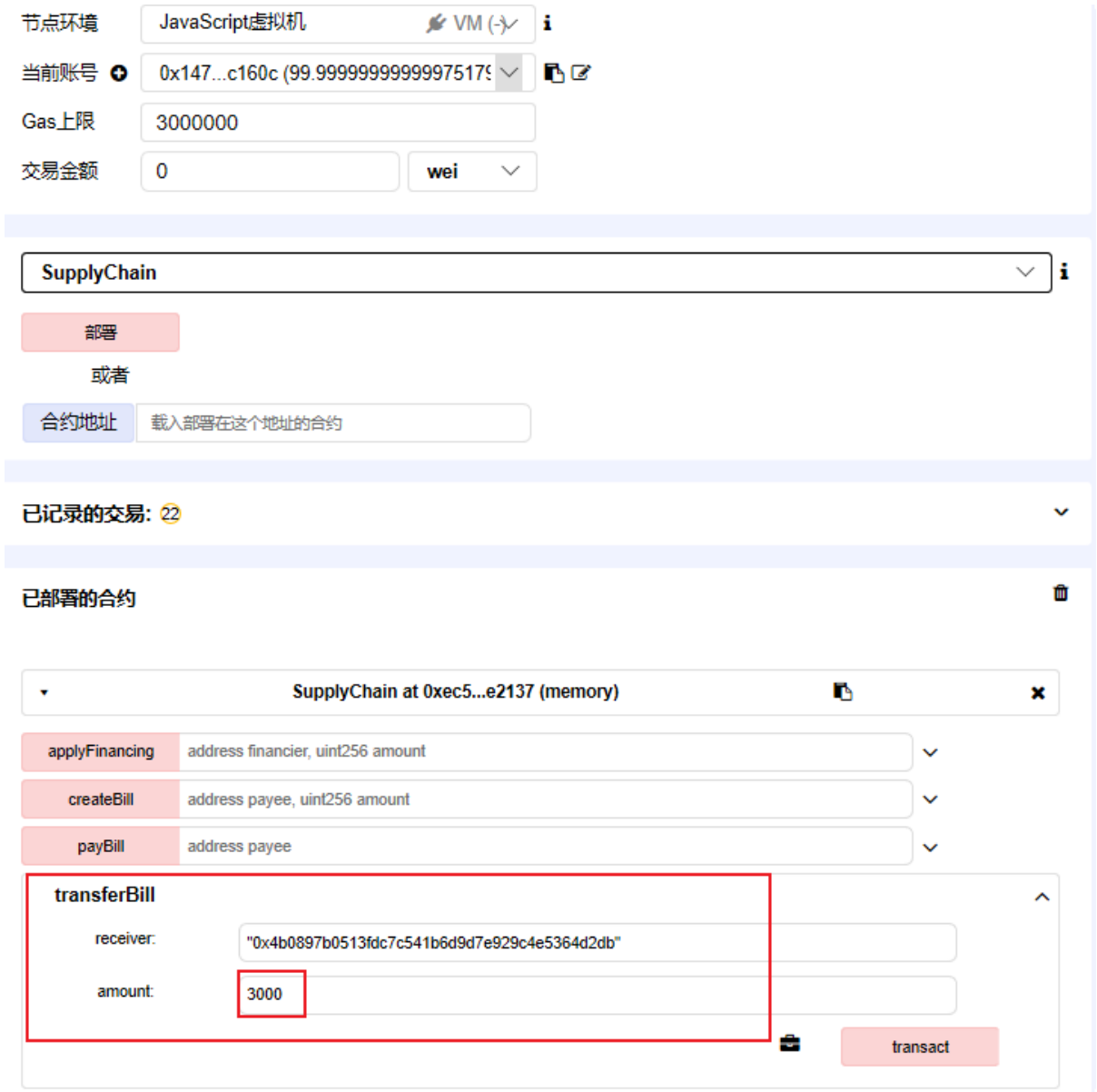
[call] from:0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db to:SupplyChain.billMap(address) data:0x91b...4d2db

Debug

transaction hash	0x5a74c1547e96a3098aad5d63ec35285122f39bb14a0cade89f61366ff5202107
from	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
to	SupplyChain.billMap(address) 0xec5bee2dbb67da8757091ad3d9526ba3ed2e2137
transaction cost	24806 gas (Cost only applies when called by a contract)
execution cost	2126 gas (Cost only applies when called by a contract)
hash	0x5a74c1547e96a3098aad5d63ec35285122f39bb14a0cade89f61366ff5202107
input	0x91b...4d2db
decoded input	{ "address ": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB" }
decoded output	{ "0": "address: payer 0x0000000000000000000000000000000000000000", "1": "address: payee 0x0000000000000000000000000000000000000000", "2": "uint256: transactionCounts 0", "3": "uint256: paymentTime 20000101", "4": "uint256: repaymentTime 20000101", "5": "uint256: billBalance 0", "6": "bool: ability false" }
logs	[]

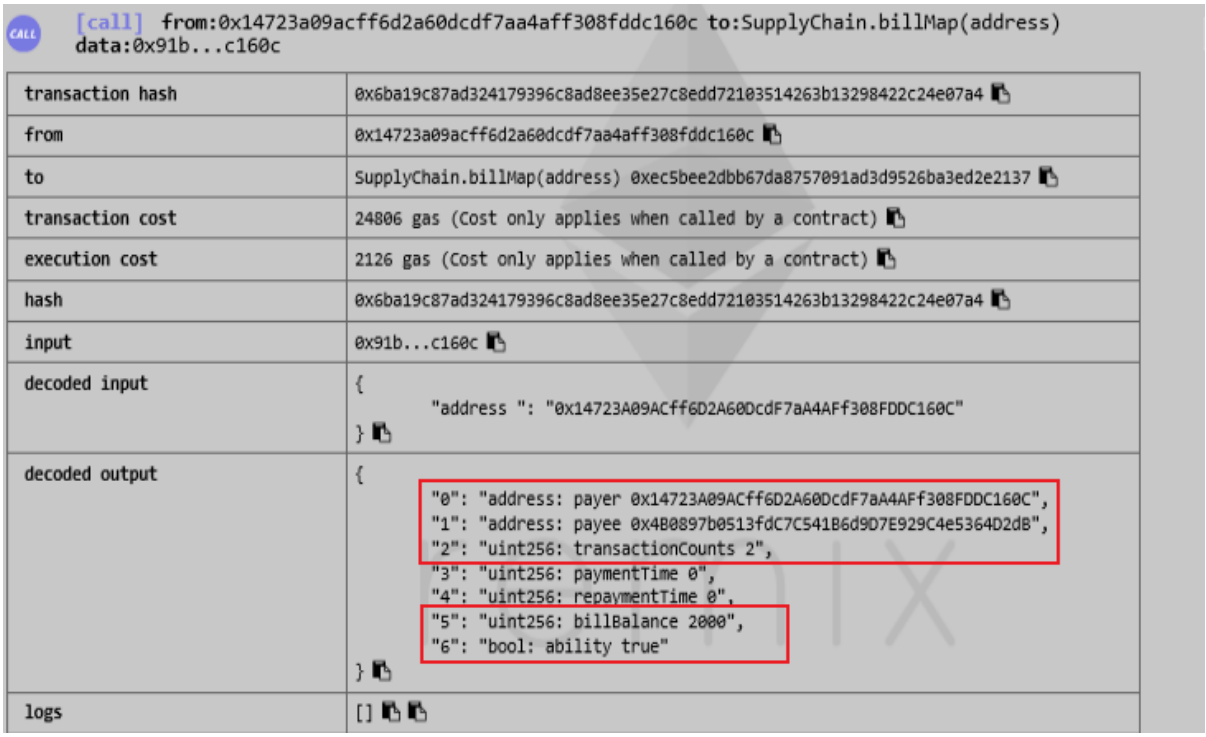
- 调用transferBill函数，当前账户为payer（Tyre），payee为Hub，amount为单据金额
  - 当payer的bill余额小于amount时结果为

- 当payer的`bill余额`不小于`amount`时结果为



decoded input	{ "address receiver": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB", "uint256 amount": "3000" }
decoded output	{}
logs	[{ "from": "0xec5bee2dbb67da8757091ad3d9526ba3ed2e2137", "topic": "0xc37c8ac0aee4f2b83144ed0c6bb3fe0f2b35eba04603af0258985c7abb2a6345", "event": "Tran_", "args": { "0": "Transfer a Bill!", "1": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "2": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB", "3": "3000", "4": "2000", "5": "3000", "6": true, "7": true, "msg": "Transfer a Bill!", "transferer": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "receiver": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB", "amount": "3000", "payerBillBalance": "2000", "payeeBillBalance": "3000", "payerBillAbility": true, "payeeBillAbility": true, "length": 8 } }]

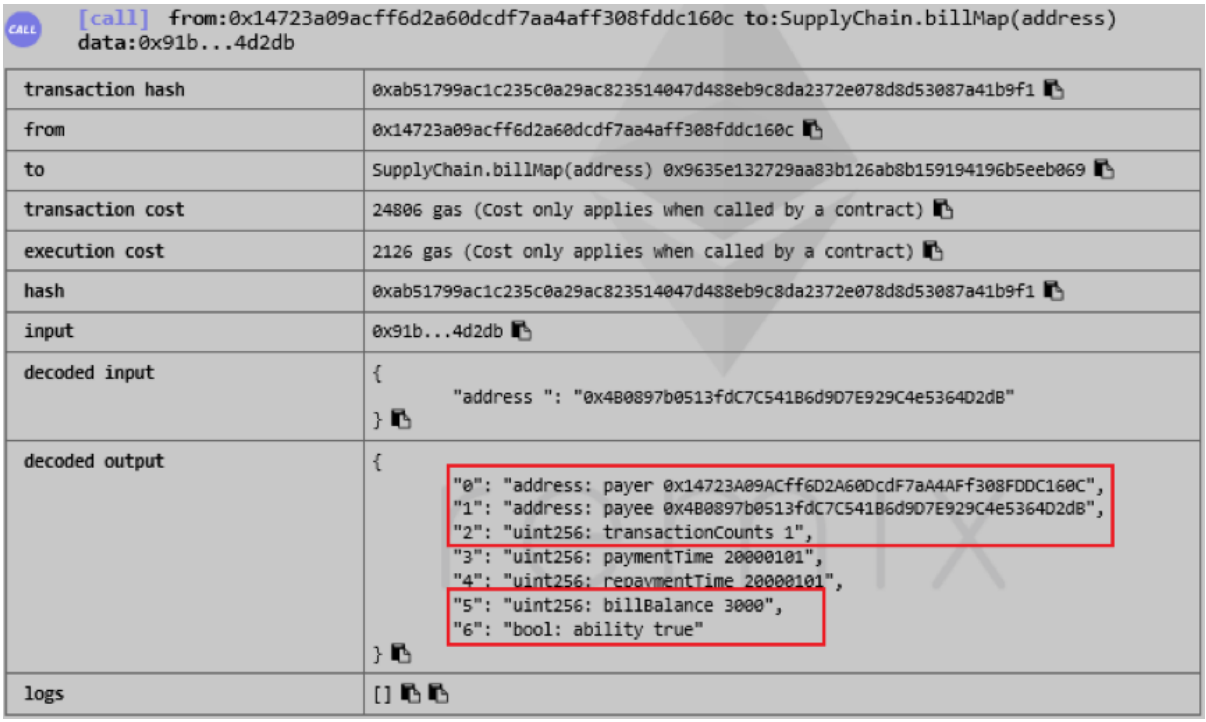
- 单据转让后的结果为：
  - Tyre公司的单据更新：payer和payee的地址更新，单据交易次数(transactionCounts)加1，billBalance减去转让的金额（5000- 3000），ability不变



[call] from:0x14723a09acff6d2a60dcdf7aa4aff308fddc160c to:SupplyChain.billMap(address) data:0x91b...c160c

transaction hash	0x6ba19c87ad324179396c8ad8ee35e27c8edd72103514263b13298422c24e07a4
from	0x14723a09acff6d2a60dcdf7aa4aff308fddc160c
to	SupplyChain.billMap(address) 0xec5bee2dbb67da8757091ad3d9526ba3ed2e2137
transaction cost	24806 gas (Cost only applies when called by a contract)
execution cost	2126 gas (Cost only applies when called by a contract)
hash	0x6ba19c87ad324179396c8ad8ee35e27c8edd72103514263b13298422c24e07a4
input	0x91b...c160c
decoded input	{ "address ": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C" }
decoded output	{ "0": "address: payer 0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "1": "address: payee 0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB", "2": "uint256: transactionCounts 2", "3": "uint256: paymentTime 0", "4": "uint256: repaymentTime 0", "5": "uint256: billBalance 2000", "6": "bool: ability true" }
logs	[]

  - Hub公司的单据更新：payer和payee的地址更新，单据交易次数(transactionCounts)加1，billBalance加上转让的金额（0 + 3000），ability为payer的ability



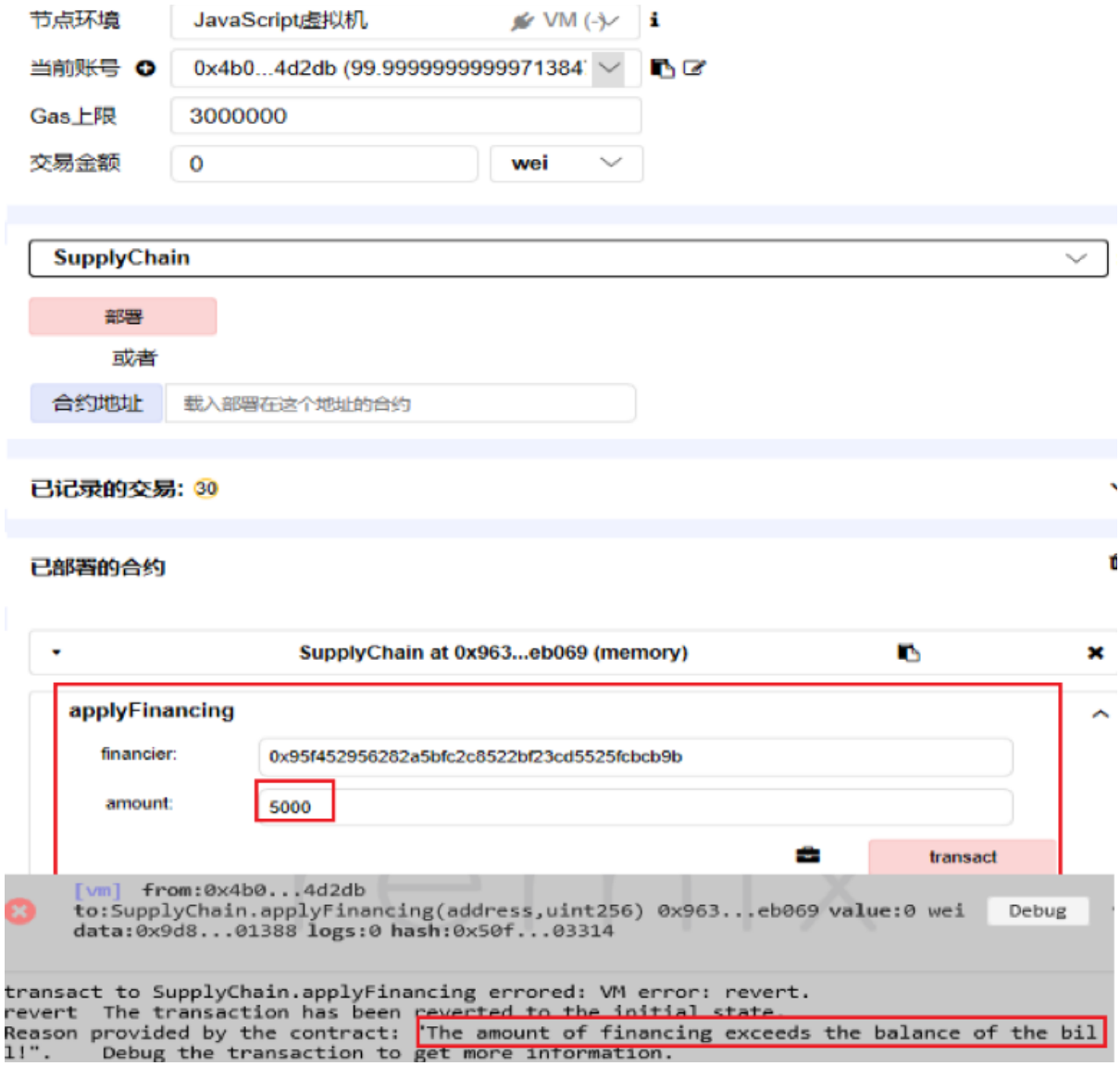
[call] from:0x14723a09acff6d2a60dcdf7aa4aff308fddc160c to:SupplyChain.billMap(address) data:0x91b...4d2db

transaction hash	0xab51799ac1c235c0a29ac823514047d488eb9c8da2372e078d8d53087a41b9f1
from	0x14723a09acff6d2a60dcdf7aa4aff308fddc160c
to	SupplyChain.billMap(address) 0x9635e132729aa83b126ab8b159194196b5eeb069
transaction cost	24806 gas (Cost only applies when called by a contract)
execution cost	2126 gas (Cost only applies when called by a contract)
hash	0xab51799ac1c235c0a29ac823514047d488eb9c8da2372e078d8d53087a41b9f1
input	0x91b...4d2db
decoded input	{ "address ": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB" }
decoded output	{ "0": "address: payer 0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "1": "address: payee 0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB", "2": "uint256: transactionCounts 1", "3": "uint256: paymentTime 20000101", "4": "uint256: repaymentTime 20000101", "5": "uint256: billBalance 3000", "6": "bool: ability true" }
logs	[]

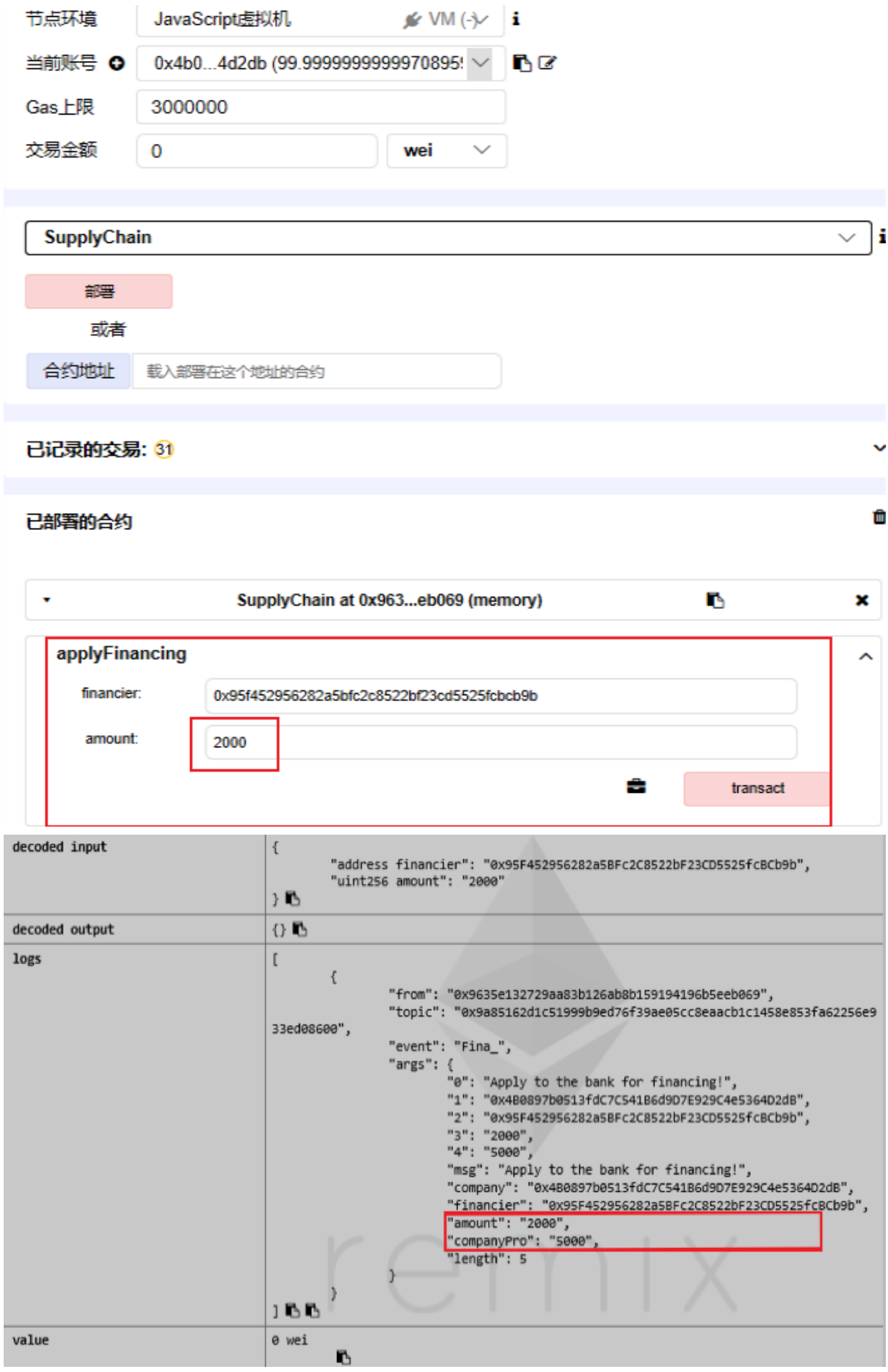
3、向银行申请融资

- 调用applyFinancing函数，Hub公司凭借单据（如上图）向银行借款（ability为true）(公司的初始property都为3000)

- 借款金额超过3000（5000），结果如下：



- 借款金额为低于3000（2000），公司的property增加2000，单据的billBalance减少2000，结果如下：

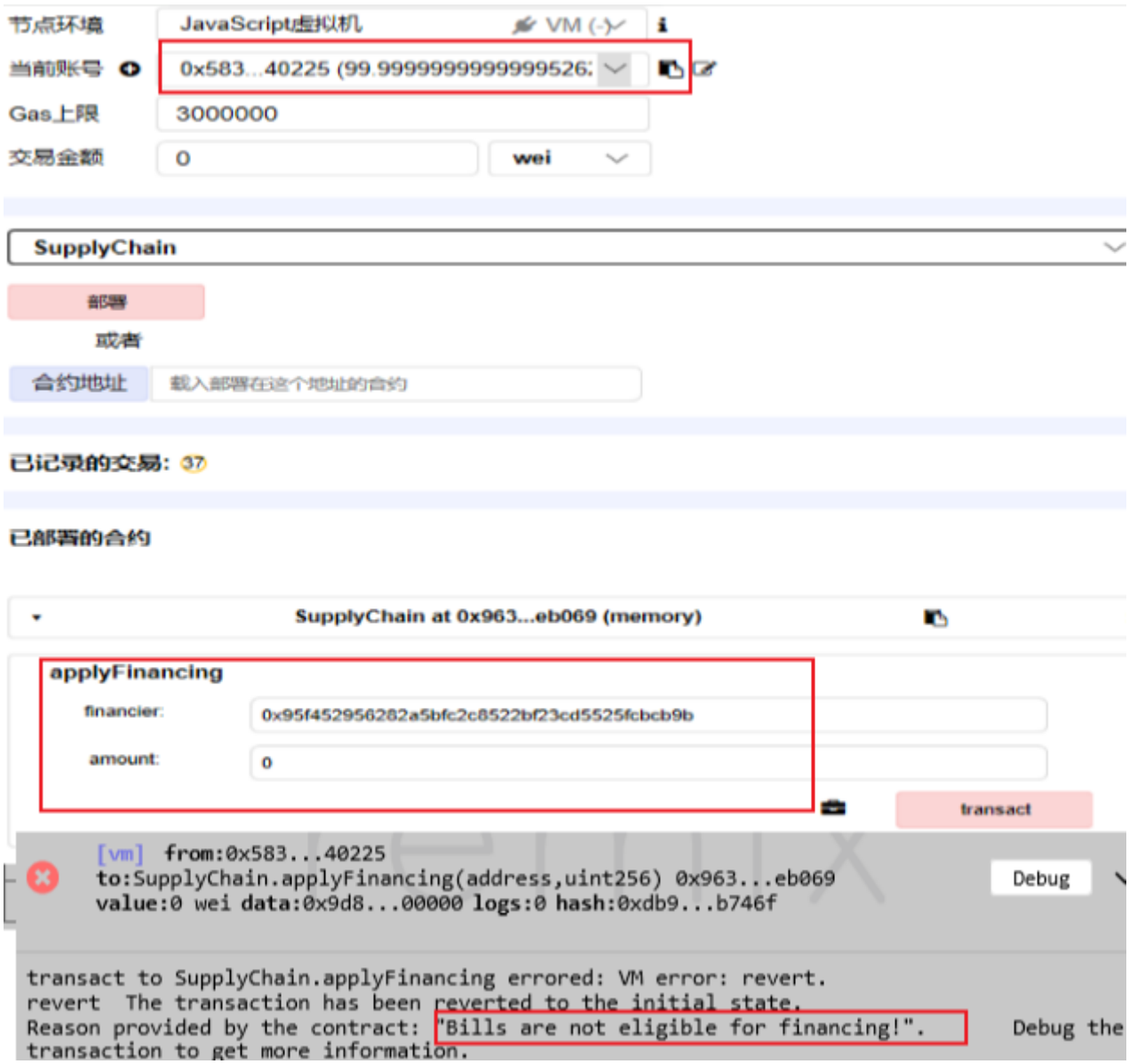

节点环境
JavaScript虚拟机
VM (-)
当前账号
0x4b0...4d2db (99.9999999999970895
Gas上限
3000000
交易金额
0
wei
SupplyChain
部署
或者
合约地址
载入部署在这个地址的合约
已记录的交易: 31
已部署的合约
SupplyChain at 0x963...eb069 (memory)
applyFinancing
financier:
0x95f452956282a5bfc2c8522bf23cd5525fcbcb9b
amount:
2000
transact
decoded input
{
"address financier": "0x95F452956282a5BFc2C8522bF23CD5525fcBCb9b",
"uint256 amount": "2000"
}
decoded output
{}
logs
[
{
"from": "0x9635e132729aa83b126ab8b159194196b5eeb069",
"topic": "0x9a85162d1c51999b9ed76f39ae05cc8eaacb1c1458e853fa62256e933ed08600",
"event": "Fina_",
"args": {
"0": "Apply to the bank for financing!",
"1": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB",
"2": "0x95F452956282a5BFc2C8522bF23CD5525fcBCb9b",
"3": "2000",
"4": "5000",
"msg": "Apply to the bank for financing!",
"company": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB",
"financier": "0x95F452956282a5BFc2C8522bF23CD5525fcBCb9b",
"amount": "2000",
"companyPro": "5000",
"length": 5
}
}
]
value
0 wei

[call] from:0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db to:SupplyChain.billMap(address) data:0x91b...4d2db

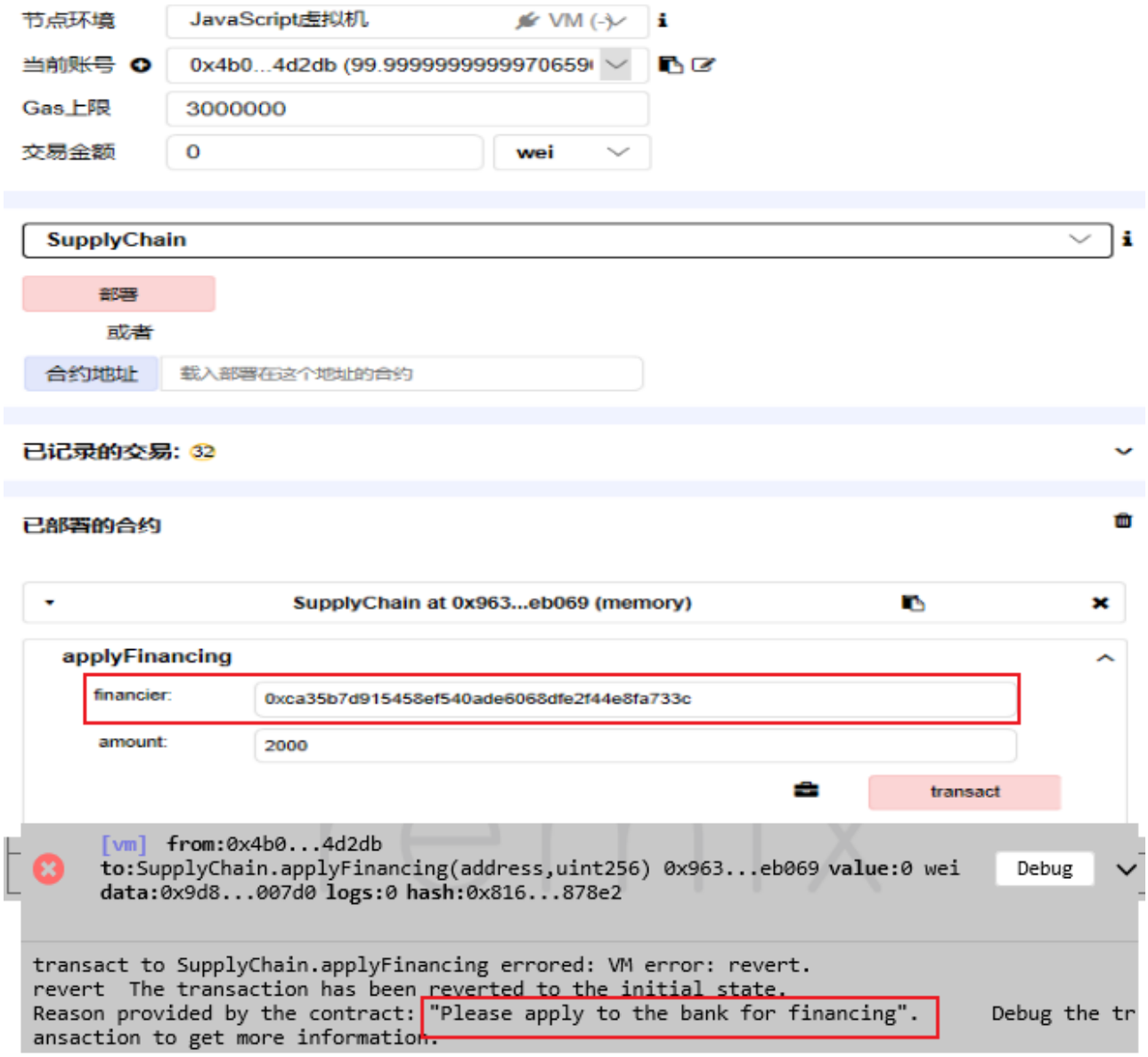
transaction hash	0xd31cd6f2a88702d14b96e1375ef2517840786c9751f8d23d2140810dbea19cef
from	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
to	SupplyChain.billMap(address) 0x9635e132729aa83b126ab8b159194196b5eeb069
transaction cost	24806 gas (Cost only applies when called by a contract)
execution cost	2126 gas (Cost only applies when called by a contract)
hash	0xd31cd6f2a88702d14b96e1375ef2517840786c9751f8d23d2140810dbea19cef
input	0x91b...4d2db
decoded input	{ "address ": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB" }
decoded output	{ "0": "address: payer 0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB", "1": "address: payee 0x95F452956282a5BFc2C8522bF23CD5525fcBCb9b", "2": "uint256: transactionCounts 1", "3": "uint256: paymentTime 20000101", "4": "uint256: repaymentTime 20000101", "5": "uint256: billBalance 1000", "6": "bool: ability true" }
logs	[]

单据的余额减2000

- Bill的ability为false时不能向Bank申请融资

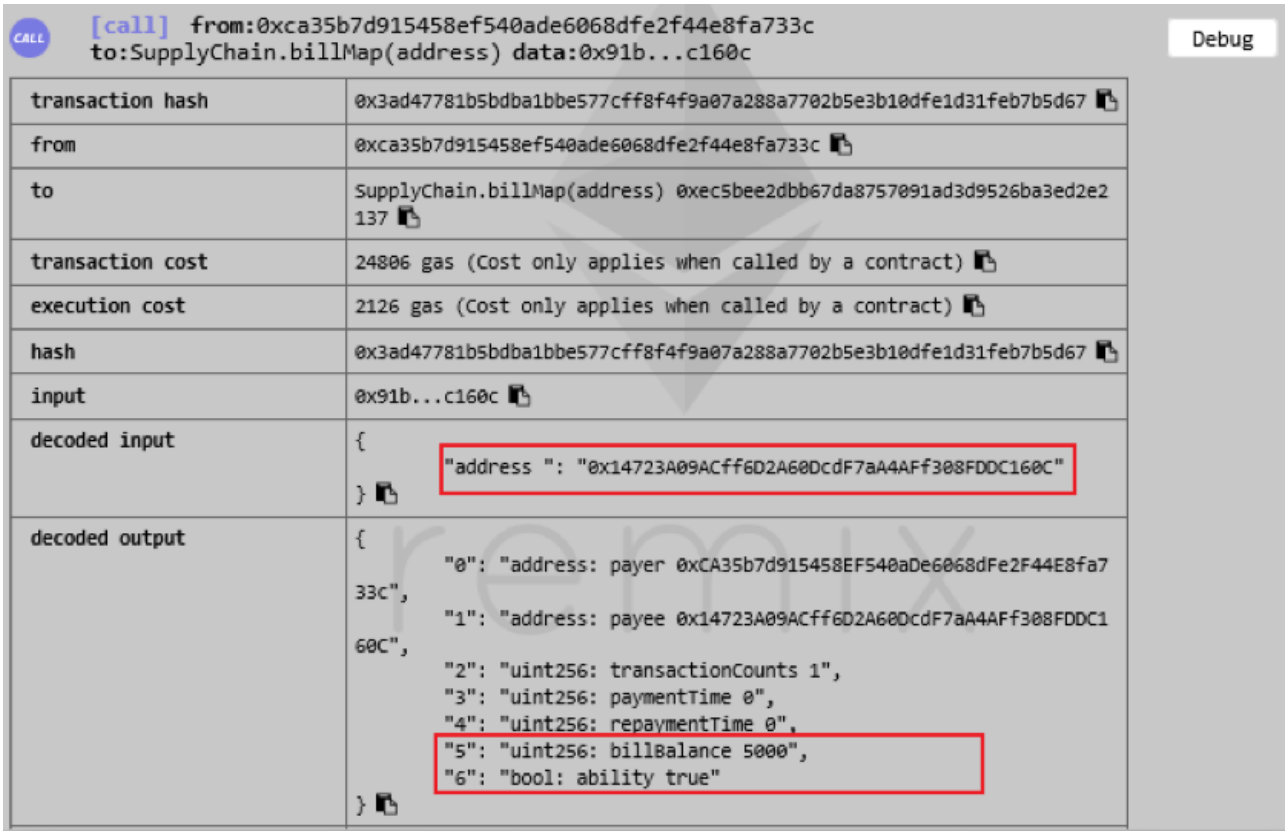


- financier不为Bank时，不能申请融资



4、单据支付结算

- 先调用billMap，查看Tyre的单据，如下：



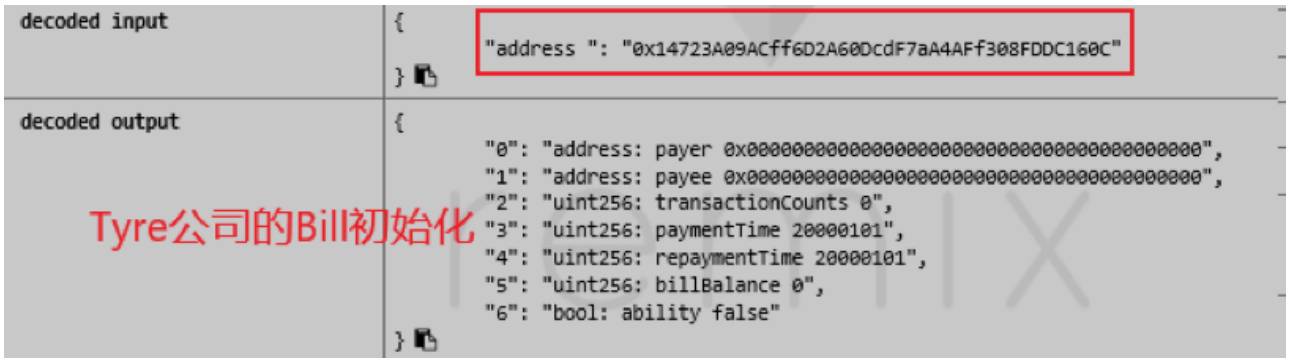
[call] from:0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to:SupplyChain.billMap(address) data:0x91b...c160c

Debug

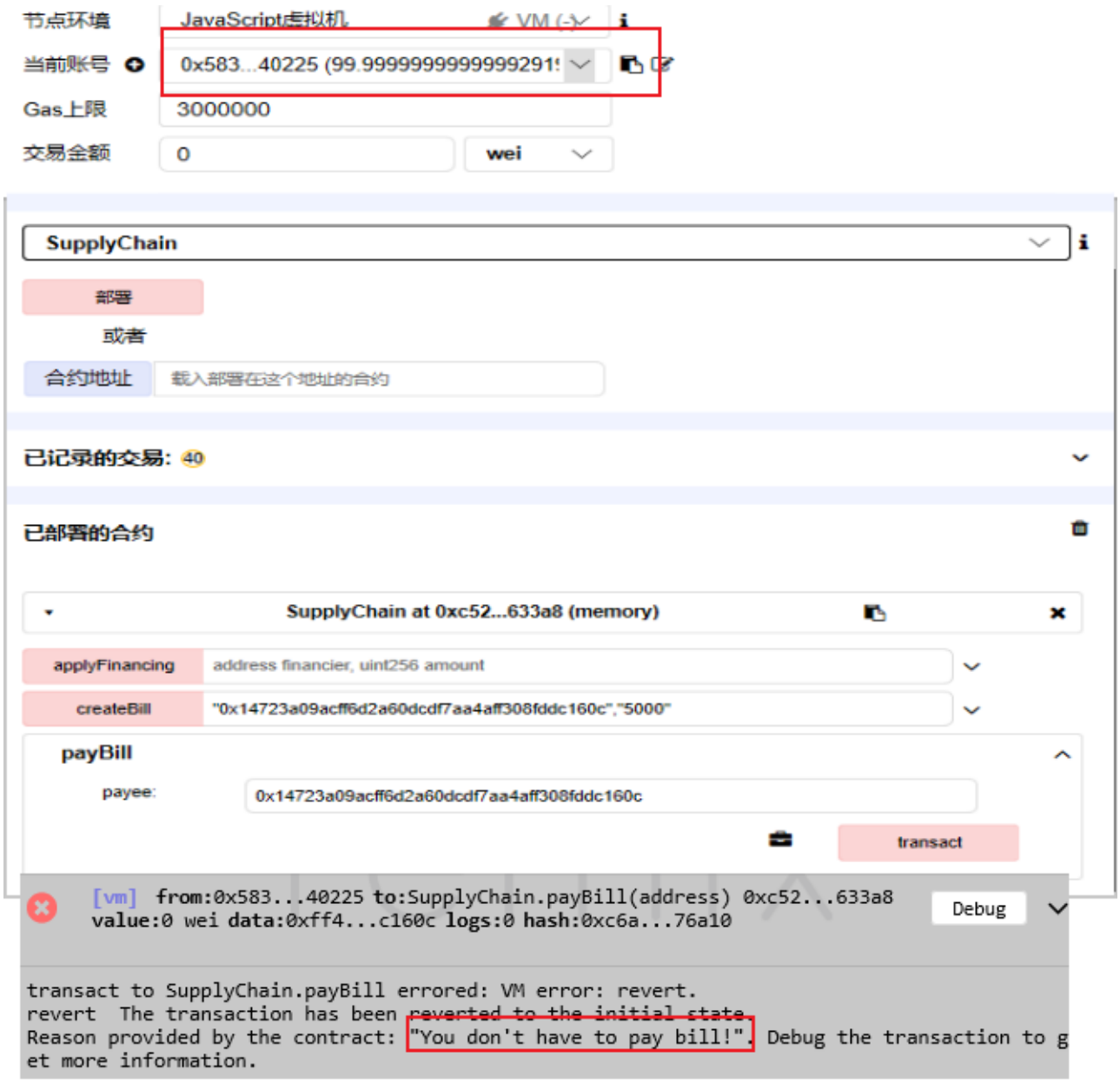
transaction hash	0x3ad47781b5bdba1bbe577cff8f4f9a07a288a7702b5e3b10dfe1d31feb7b5d67
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	SupplyChain.billMap(address) 0xec5bee2dbb67da8757091ad3d9526ba3ed2e2137
transaction cost	24806 gas (Cost only applies when called by a contract)
execution cost	2126 gas (Cost only applies when called by a contract)
hash	0x3ad47781b5bdba1bbe577cff8f4f9a07a288a7702b5e3b10dfe1d31feb7b5d67
input	0x91b...c160c
decoded input	{ "address ": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C" }
decoded output	{ "0": "address: payer 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c", "1": "address: payee 0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "2": "uint256: transactionCounts 1", "3": "uint256: paymentTime 0", "4": "uint256: repaymentTime 0", "5": "uint256: billBalance 5000", "6": "bool: ability true" }

- Car调用payBill函数，payee为Tyre，Car的property减去Tyre的bill的billBalance（10000 - 5000 = 5000），Tyre增加相应金额(3000 + 5000 = 8000)，然后Tyre的bill被重新初始化

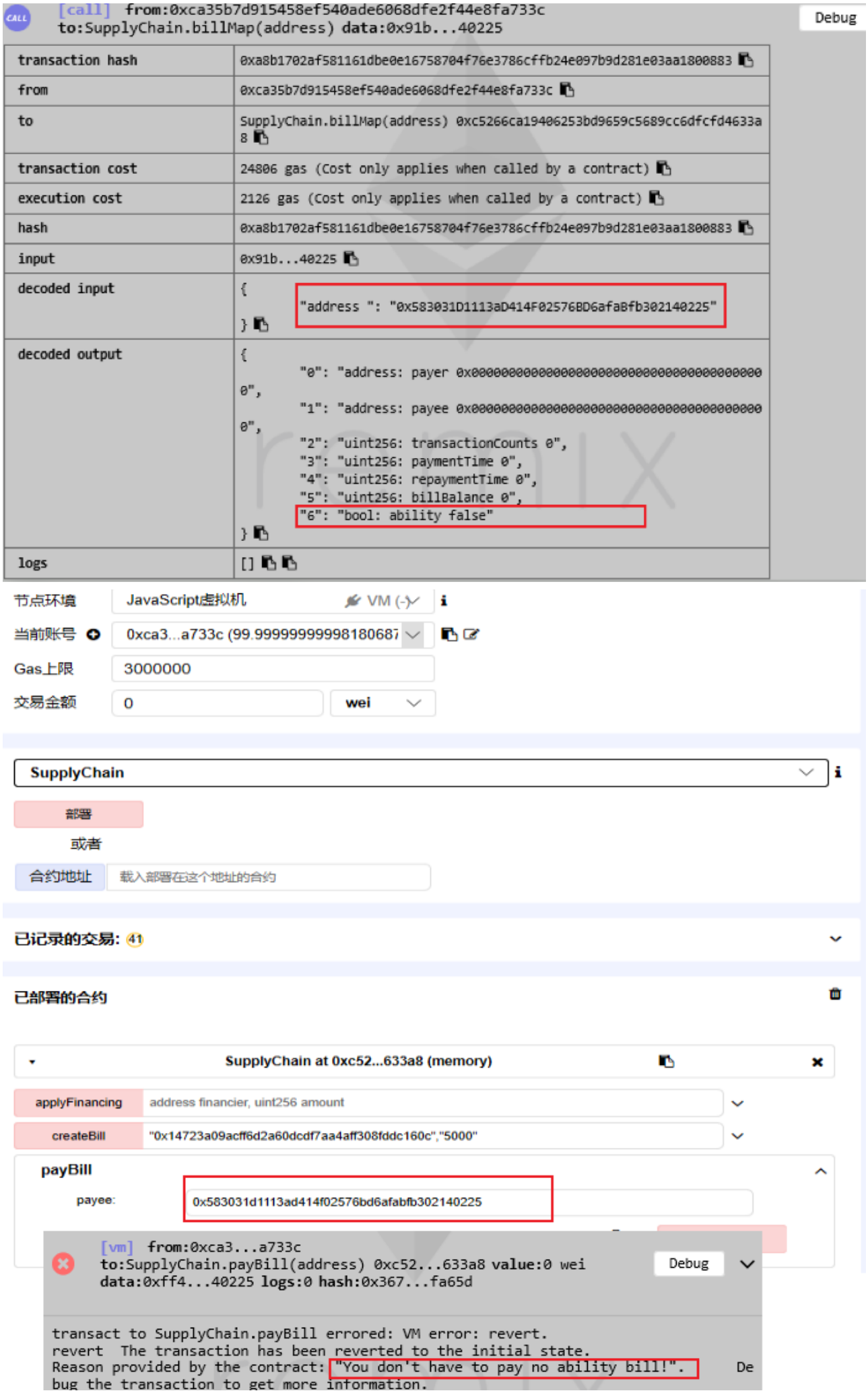
decoded input	{ "address payee": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C" }
decoded output	{}
logs	[{ "from": "0xc5266ca19406253bd9659c5689cc6dfcfd4633a8", "topic": "0x4e91bfc266d06b2bcd93d298c500ed8ba0c7e7e05d037cc624ff933dd7624de4", "event": "Pay_", "args": { "0": "Pay a bill!", "1": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c", "2": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "3": "5000", "4": "5000", "5": "8000", "msg": "Pay a bill!", "payer": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c", "payee": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C", "amount": "5000", "payerPro": "5000", "payeePro": "8000", "length": 6 } }]
value	0 wei



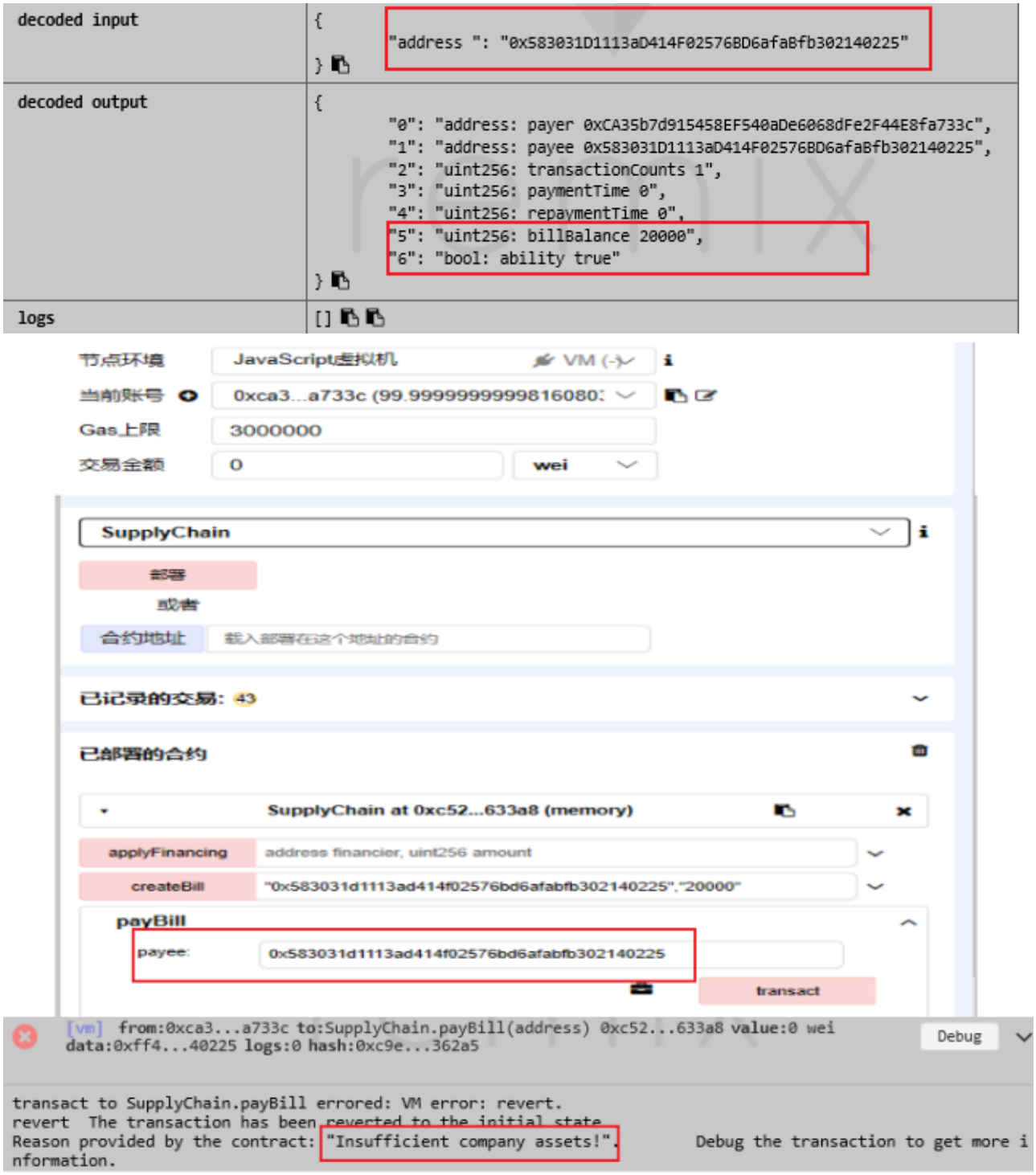
- 非Car公司不能调用此函数



- bill的ability为false时不用还款


[call] from:0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to:SupplyChain.billMap(address) data:0x91b...40225
Debug
transaction hash
0xa8b1702af581161dbe0e16758704f76e3786cffb24e097b9d281e03aa1800883
from
0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to
SupplyChain.billMap(address) 0xc5266ca19406253bd9659c5689cc6dfcfd4633a8
transaction cost
24806 gas (Cost only applies when called by a contract)
execution cost
2126 gas (Cost only applies when called by a contract)
hash
0xa8b1702af581161dbe0e16758704f76e3786cffb24e097b9d281e03aa1800883
input
0x91b...40225
decoded input
{
"address ": "0x583031D1113aD414F02576BD6afaBfb302140225"
}
decoded output
{
"0": "address: payer 0x0000000000000000000000000000000000000000",
"1": "address: payee 0x0000000000000000000000000000000000000000",
"2": "uint256: transactionCounts 0",
"3": "uint256: paymentTime 0",
"4": "uint256: repaymentTime 0",
"5": "uint256: billBalance 0",
"6": "bool: ability false"
}
logs
[]
节点环境
JavaScript虚拟机
VM (-)
当前账号
0xca3...a733c (99.99999999998180687
Gas上限
3000000
交易金额
0
wei
SupplyChain
部署
或者
合约地址
载入部署在这个地址的合约
已记录的交易: 41
已部署的合约
SupplyChain at 0xc52...633a8 (memory)
applyFinancing
address financier, uint256 amount
createBill
"0x14723a09acff6d2a60dcdf7aa4aff308fddc160c","5000"
payBill
payee:
0x583031d1113ad414f02576bd6afabfb302140225
[vm] from:0xca3...a733c
to:SupplyChain.payBill(address) 0xc52...633a8 value:0 wei
data:0xff4...40225 logs:0 hash:0x367...fa65d
Debug
transact to SupplyChain.payBill errored: VM error: revert.
revert The transaction has been reverted to the initial state.
Reason provided by the contract: "You don't have to pay no ability bill!".
De
bug the transaction to get more information.

- 公司property不足时不能结算



四、后端（Java）

通过FISCO BCOS控制台提供的方法先将合约转换为Java类(源码地址)：

- 将合约文件放入fisco-bcos/console/contracts/solidity下

```
lucky@ubuntu:~/fisco-bcos/console/contracts/solidity$ ls
SupplyChain.sol
```

- 在console目录下指定包名运行sol2java.sh脚本

```
$ cd ~/fisco/console
$ ./sol2java.sh org.com.fisco
```

```
lucky@ubuntu:~/fisco-bcos/console$ ./sol2java.sh org.com.fisco

Compile solidity contract files to java contract files successfully!
lucky@ubuntu:~/fisco-bcos/console$
```

- 运行成功之后，将会在console/contracts/sdk目录生成java、abi和bin目录，如下所示：

```
lucky@ubuntu:~/fisco-bcos/console/contracts/sdk$ tree
.
├── abi
│   └── SupplyChain.abi
├── bin
│   └── SupplyChain.bin
└── java
    └── org
        └── com
            └── fisco
                └── SupplyChain.java

6 directories, 3 files
lucky@ubuntu:~/fisco-bcos/console/contracts/sdk$
```

- 在fisco-bcos上部署SupplyChain.sol

```
lucky@ubuntu:~/fisco-bcos/console$ ./start.sh
=====================================================================================================
Welcome to FISCO BCOS console(1.0.5)!
Type 'help' or 'h' for help. Type 'quit' or 'q' to quit console.
 ________ ______  ______   ______   ______       _______   ______   ______   ______
|        |      \/      \ /      \ /      \     |       \ /      \ /      \ /      \
| $$$$$$$$\$$$$$|  $$$$$$|  $$$$$$|  $$$$$$\    | $$$$$$$|  $$$$$$|  $$$$$$|  $$$$$$\
| $$__     | $$ | $$___\$| $$   \$| $$  | $$    | $$__/ $| $$   \$| $$  | $| $$___\$$
| $$  \    | $$  \$$    \| $$      | $$  | $$    | $$    $| $$      | $$  | $$\$$    \
| $$$$$    | $$  _\$$$$$$| $$   __| $$  | $$    | $$$$$$$| $$   __| $$  | $$_\$$$$$$\
| $$      _| $$_|  \__| $| $$__/  | $$__/ $$    | $$__/ $| $$__/  | $$__/ $|  \__| $$
| $$     |   $$ \\$$    $$\$$    $$\$$    $$    | $$    $$\$$    $$\$$    $$\$$    $$
 \$$      \$$$$$$ \$$$$$$  \$$$$$$  \$$$$$$      \$$$$$$$  \$$$$$$  \$$$$$$  \$$$$$$

=====================================================================================================
[group:1]> deploy SupplyChain.sol
contract address: 0x6dbcb95d8c4416bcf49b53539a9395d5c6821109

[group:1]>
```

五、前端界面展示（VUE.js）

前端使用vue框架搭建，npm安装框架

- 首先安装npm：sudo apt install npm(依赖nodejs，所以需要先安装node)
- 然后安装vue-cli脚手架：sudo npm install -g vue-cli（-g为安装到全局）
  - node和npm版本要在4以上（node -v, npm -v）
  - 由于最初的源使用npm install命令时非常的慢，所以可以通过命令npm config set registry https://registry.npm.taobao.org切换阿里的源
  - 通过npm config get registry查看npm的源

- 然后进入vue项目中（含vue的目录结构），通过命令npm install安装依赖
- 通过命令npm run dev在本地端口8080启动项目

  ……

六、心得体会

这次作业虽然真正只实现了链端，但收获还是蛮多的，学习了一门新的语言solidity，在有其他语言的基础下，solidity的学习还是比较容易的，只是要注意一些特别的变量和函数而已，像address、bytes、event、emit等。除此之外，还通过使用FISCO BCOS和WEBASE了解了区块链链端的知识，包括链的搭建，合约的部署以及合约的调用等等。