

区块链大作业：基于区块链的供应链金融平台

功能测试文档

区块链大作业：基于区块链的供应链金融平台

功能测试文档

- 一、文档测试内容（主要测试 **项目设计说明** 中设计的功能部分）
- 二、测试交易例子
- 三、功能测试
 1. 初始化部署合约
 2. 四个基础功能测试
 - ① 注册功能 (register)
 - ② 查询账户信息功能 (selectAccount)
 - ③ 查询交易信息功能 (selectReceipt)
 - ④ 资产转移功能 (transfer)
 3. 四个基本交易功能测试
 - ① 签发应收账款交易上链（添加交易 createReceipt）
 - ② 应收账款的转让上链（拆分交易 divideReceipt）
 - ③ 利用应收账款向银行融资上链（融资 financeReceipt）
 - ④ 应收账款支付结算上链（结算交易 repayReceipt）

课程名称	区块链原理与技术	任课老师	黄华威
年级	2018级	专业	软件工程
小组	Group 6	大作业	基于区块链的供应链金融平台
组员	黄进，胡梓渊，喻勇强	学号	18342029，18342026，18342123
开始日期	2020.11.30	完成日期	2020.12.14

本次链端智能合约的编写主要参考了 **FISCO BCOS** 官方文档，参考的教程文档如下：

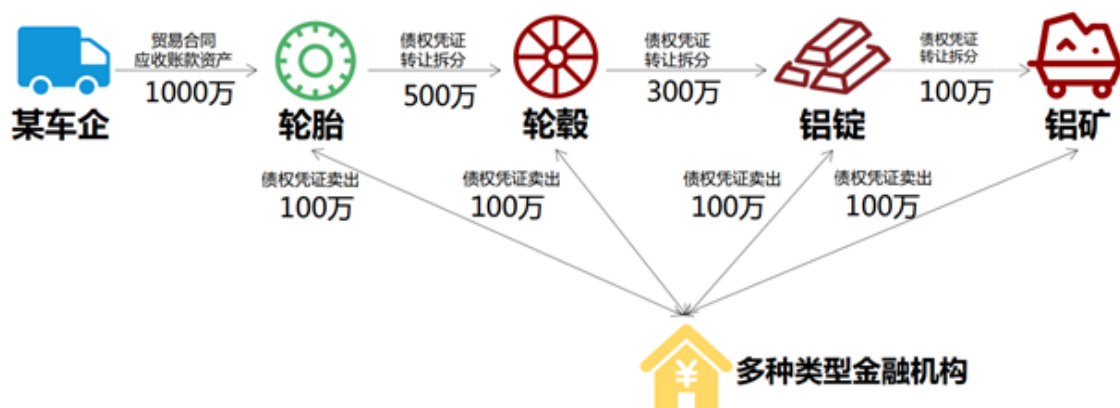
- [构建第一个区块链应用](#)
- [FISCO BCOS零基础入门，五步轻松构建应用](#)
- [智能合约编写之Solidity的基础特性](#)

一、文档测试内容（主要测试 **项目设计说明** 中设计的功能部分）

- 将智能合约部署至链上（多节点），并调用相关函数，详细说明上述的四个功能具体是如何实现的
- 要测试的四个功能有：
 1. 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。
 2. 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。
 3. 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。
 4. 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、测试交易例子

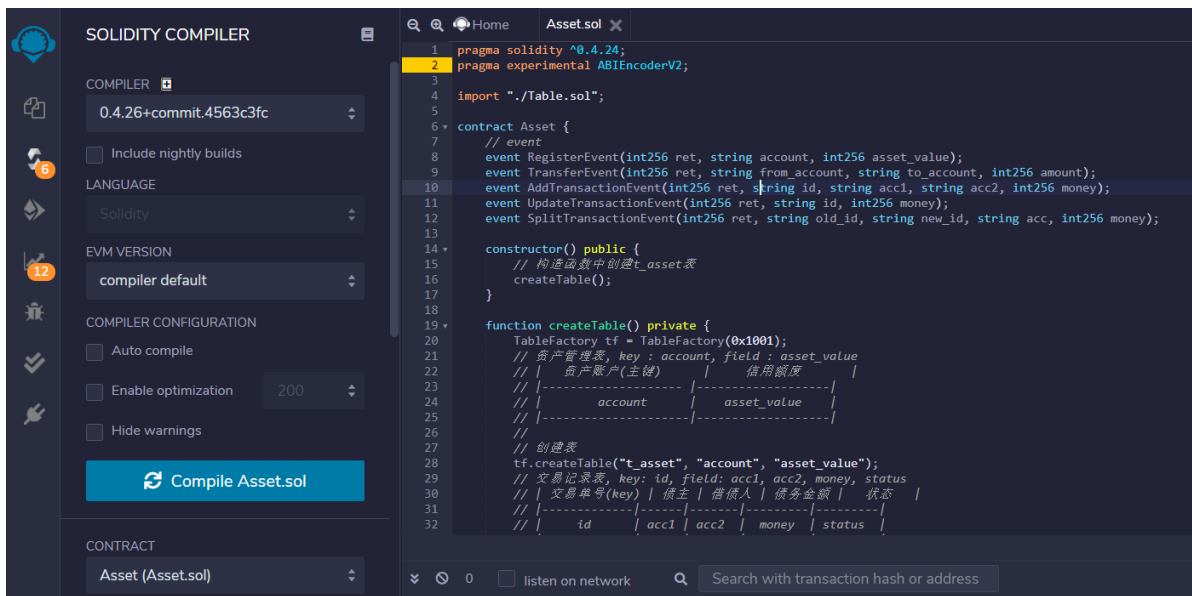
- 在这次的项目中，需要用到的在链上的对象一共有四个，分别是上游企业车企、中游轮胎公司、下游轮胎公司和银行。
- 合约实现以下四个交易功能，即添加交易上链、交易转让上链、融资上链和支付结算上链
- 以车企，轮胎厂，轮胎厂，和银行间的交易作为例子：它们的资产额度假设分别为 A_0 , B_0 , C_0 , D_0 ，注意银行的资产额度非常大



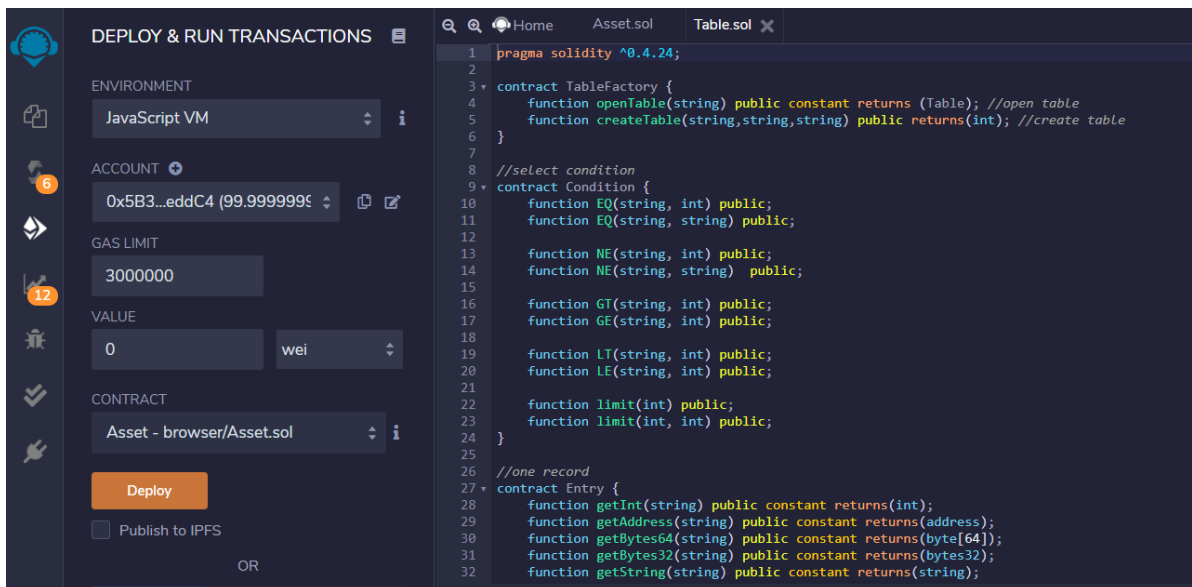
1. 某上游企业车企与中游企业轮胎厂进行一笔交易，交易金额为1000万，此时通过 `createReceipt` 函数创建一笔交易，交易过后，车企的资产额度变为 $A_1 = A_0 - 1000$ ，轮胎厂的变为 $B_1 = B_0 + 1000$
2. 轮胎厂通过拆分转让和车企的交易，与轮胎厂产生一笔新的交易，交易金额为500万，通过 `divideReceipt` 来实现。这时轮胎厂的资产额度变为 $B_2 = B_1 - 500$ ，轮胎厂变为 $C_1 = C_0 + 500$
3. 轮胎厂再向银行融资500万 `financeReceipt`，此时轮胎厂资产额度变为 $B_3 = B_2 - 500$ ，银行变为 $D_1 = D_0 + 500$
4. 车企向轮胎厂还款 `repayReceipt`，还款金额为500万，车企的资产额度变为 $A_2 = A_1 + 500$ ，轮胎厂 $B_4 = B_3 - 500$

三、功能测试

- 运行合约的方式有多种，比如
 1. 方法一：可以使用[FISCO BCOS控制台](#)的方式来部署合约
 2. 方法二：使用FISCO BCOS开源项目WeBASE提供的在线ide WeBASE-front运行
 3. 方法三：通过在线ide remix来进行合约的部署与运行, [remix的地址](#)
- 以下是通过 remix ide 编译部署合约的过程



- 编译成功后就可进行部署环节，部署成功后会出现合约实例



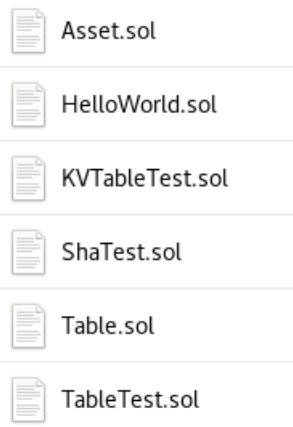
- 综合考虑过后，下文我们小组将使用 **FISCO BCOS 控制台** 的方式来部署合约并测试功能

1. 初始化部署合约

- 首先启动所有节点，检查连接和共识

```
1 // 检查进程是否启动
2 ps -ef | grep -v grep | grep fisco-bcos
3 // 检查日志输出，查看节点 newNode 链接的节点数
4 tail -f log/log* | grep connected
5 // 检查是否在共识，不停输出 ++++Generating seal，表示共识正常
6 tail -f log/log* | grep +++
```

- 然后启动并使用 **FISCO BCOS** 控制台

[illegible]

```
[group:1] > deploy Asset
transaction hash: 0x52647dc312837f9a18b839f032c8741e7a4d7d71ebe1929b112b36e37434fa43
contract address: 0xa7820227266c7ea6df8770d56bde68b52d0ca022
```

2. 四个基础功能测试

① 注册功能 (register)

- 注册添加四个用户，分别为车企 (car)，轮胎厂 (wheel)，轮毂厂 (hub)，和第三方权威机构银行 (bank) 间的交易作为例子：它们的资产额度分别假设为 9999, 2000, 999, 99999

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 register car 9999
transaction hash: 0xa7d295e28aa3b4f229432ccb598a3ec453e0c3eeel798a4d0e6e30ca7ada61f1
```

```
-----
transaction status: 0x0
description: transaction executed successfully
-----
```

```
Output
Receipt message: Success
Return message: Success
Return value: [0]
```

```
-----
Event logs
```

```
Event: {"RegisterEvent": [[0, "car", 9999]]} 注册结果，注册的用户名，资产额度
```

② 查询账户信息功能 (selectAccount)

- 根据账户名查询一个账户是否存在，它的资产金额是多少。比如查询车企公司的资产金额，为9999

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectAccount car
```

```
-----
Return code: 0
description: transaction executed successfully
Return message: Success
-----
```

```
Return values:
```

```
[
  [0, 9999] 查询结果，账户资产金额
]
```

③ 查询交易信息功能 (selectReceipt)

- 根据交易编号 ID 查询一笔交易是否存在，它的交易金额和未结清金额是多少，债权人和借债人是谁。比如此前的 ID 为 18 的交易是一笔借债人为“yyq”，债权人为“hj”，交易金额为 100，未结算金额为0，即已完成的交易订单

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectReceipt 18
```

```
-----
Return code: 0
description: transaction executed successfully
Return message: Success
-----
```

```
Return values:
```

```
[
  0,
  [
    100, 交易金额，未结算金额
    0
  ],
  [
    "yyq", 借债人，债权人
    "hj"
  ]
]
```

交易编号

④ 资产转移功能 (transfer)

- 将金额从一个账户转移到另一个，比如从银行 bank 转移 1000 的资产金额给车企 car

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 transfer bank car 1000
transaction hash: 0x071ba4d4b5d3f4cd4629a9d5028e0768a3f5defb7bf488f5d0a06a605492562c
```

```
-----
transaction status: 0x0
description: transaction executed successfully
-----
```

Output

Receipt message: Success

Return message: Success

Return value: [0]

Event logs

Event: {"TransferEvent": [0, "bank", "car", 1000]}

从谁 到谁 转移资产金额

- 重新查询车企的资产金额，增加了 1000 变为 10999

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectAccount car
```

```
-----
Return code: 0
```

description: transaction executed successfully

Return message: Success

```
-----
Return values:
```

```
[
```

```
0,
10999
```

9999 + 1000 -> 10999

```
]
```

3. 四个基本交易功能测试

① 签发应收账款交易上链 (添加交易 createReceipt)

- 首先是验证功能一，createReceipt 通过调用资产转移函数 transfer 实现创建新交易功能
- 只有上游企业（资产额度足够大）才能发起交易，上游企业车企 car 与中游企业 wheel 进行交易，car 欠下 wheel 1000 资金，两者之间签订了一笔交易，为 [id25, car, wheel, 1000, 1000]

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 createReceipt 25 car wheel 1000
transaction hash: 0x4106e9e7ac2820da7d3dc3fd5fa2e3186fba3bf69d94ac4030feb7dfc97816b7
```

```
-----
transaction status: 0x0
description: transaction executed successfully
-----
```

Output

Receipt message: Success

Return message: Success

Return value: [0]

Event logs

Event: {"TransferEvent": [0, "car", "wheel", 1000], "CreateReceiptEvent": [0, "25", "car", "wheel", 1000]}

- 车企 car 的资产额度变为 $10999 - 1000 = 9999$ ，轮胎厂 wheel 的变为 $2000 + 1000 = 3000$ ，使用 selectAccount 来查看两者的资产金额

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectAccount car
```

```
Return code: 0
description: transaction executed successfully
Return message: Success
```

Return values:

```
[
  0,
  9999
]
```

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectAccount wheel
```

```
Return code: 0
description: transaction executed successfully
Return message: Success
```

Return values:

```
[
  0,
  3000
]
```

- 功能一 签发应收账款交易上链 验证完毕

② 应收账款的转让上链 (拆分交易 divideReceipt)

- 接下来验证功能二，假设一个用户是一笔交易的债权人，现在他想转让到另一笔向别的用户借债的新交易中，可以将不超过原交易未结清金额的额度转移给另外一个用户
- 通过转让原交易借债人的应收账款给新交易的债权人，实现交易转移，新债权人会从该用户那获得同等额度的资产
- debtee 把 borrower 借的债转让给 newDebtee，debtee 资产额度转移给 newDebtee，相当于分为两步
 1. borrower 还债给 debtee (repayReceipt)
 2. borrower 向 newDebtee 借债 (createReceipt)
- 调用 divideReceipt，让轮胎厂 wheel 向 轮毂厂 hub 借 500 资金；由于轮胎厂 wheel 在前面已有了一条借给车企 car 1000 资金的应收账款，它可转让这笔应收账款。

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 divideReceipt 25 26 hub 500
transaction hash: 0xcd12ec2d7df2038e7d439942d1c37e028abb2e4ff4a5b4387d7f610c7178c849
.....
transaction status: 0x0
description: transaction executed successfully
.....
Output
Receipt message: Success
Return message: Success
Return value: [0]
.....
Event logs
Event: [{"DivideReceiptEvent": [{"id": "25", "debtor": "hub", "amount": 500}], "TransferEvent": [{"id": "wheel", "debtor": "car", "amount": 500}, {"id": "car", "debtor": "hub", "amount": 500}], "CreateReceiptEvent": [{"id": "26", "debtor": "car", "amount": 500}], "RepayReceiptEvent": [{"id": "25", "debtor": "hub", "amount": 500}]}]
```

- 此时上一条交易变为 [id25, 车企, 轮胎厂, 1000, 500]，新增一条交易 [id26, 车企, 轮毂厂, 500, 500]


```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectReceipt 25
```

```
Return code: 0
description: transaction executed successfully
Return message: Success
```

```
Return values:
```

```
[
  0,
  [
    1000,
    500
  ],
  [
    "car",
    "wheel"
  ]
]
```

- 注意 新交易26 里，借债人是车企 car 而不是轮胎厂 wheel

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectReceipt 26
```

```
Return code: 0
description: transaction executed successfully
Return message: Success
```

```
Return values:
```

```
[
  0,
  [
    500,
    500
  ],
  [
    "car",
    "hub"
  ]
]
```

- 查看轮胎厂 wheel 的资产额度，更新为 $3000 - 500 = 2500$ ，轮毂厂 hub 资产额度变为 $999 + 500 = 1499$
- 使用 `selectAccount` 来查看两者的资产金额

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectAccount wheel
```

```
Return code: 0
description: transaction executed successfully
Return message: Success
```

```
Return values:
```

```
[
  0,
  2500
]
```

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectAccount hub
```

```
Return code: 0
description: transaction executed successfully
Return message: Success
```

```
Return values:
```

```
[
  0,
  1499
]
```

- 至此功能二 应收账款的转让上链 验证完毕

③ 利用应收账款向银行融资上链 (融资 financeReceipt)

- 接下来是功能三的验证, 把应收账款转让给银行, 从而从银行获得融资
- 所有企业都可以使用自身的资产额度向银行请求融资, 获得一笔不多于自身资产额度的金额。
- 调用 `financeReceipt`, 把轮胎厂 wheel 的资产额度转让给银行获得融资

```
[group:1]> call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 financeReceipt 27 wheel 500 交易编号, 融资人, 融资金额
transaction hash: 0xecee81a42179506d337b772f5a893ff79d0c6ba66294d67b76a2419760527160
-----
transaction status: 0x0
description: transaction executed successfully
-----
Output
Receipt message: Success
Return message: Success
Return value: [0]
-----
Event logs
Event: {"TransferEvent": [[0, "wheel", "bank", 500]], "CreateReceiptEvent": [[0, "27", "wheel", "bank", 500]], "FinanceReceiptEvent": [[0, "27", "wheel", 500]]}
```

- 创建了一单交易 [id27, 轮胎厂, 银行, 500, 500] (参数只有融资人, 默认债权人是银行 bank)

```
[group:1]> call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectReceipt 27
-----
Return code: 0
description: transaction executed successfully
Return message: Success
-----
Return values:
[
  [
    0,
    [
      500,
      500
    ],
    [
      "wheel",
      "bank"
    ]
  ]
]
```

- 查看轮胎厂 wheel 的资产额度, 更新为 $2500 - 500 = 2000$

```
[group:1]> call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectAccount wheel
-----
Return code: 0
description: transaction executed successfully
Return message: Success
-----
Return values:
[
  [
    0,
    2000
  ]
]
```

- 至此功能三 利用应收账款向银行融资上链 验证完毕。

④ 应收账款支付结算上链 (结算交易 repayReceipt)

- 最后是功能四的验证, 通过资产转移来移除借款人手中的应收账款, 借债人和债权人资产额度变化结算的金额
- 在一笔交易中, 作为借债人的企业可以通过还钱给债主来让交易中的未结清金额减少, 并从债权人那返还等额的资产额度。当未结清金额为 0 时, 该交易支付结算完毕
- 调用 `repayReceipt`, 以车企 car 还款给轮胎厂 wheel 为例, 假设还款金额为 500。则 交易25 已还清

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 repayReceipt 25 500
transaction hash: 0x62739b42c26de58f4e0f061309391cbb6262c35673780921f10d8f8e61f83c2a
-----
transaction status: 0x0
description: transaction executed successfully
-----
Output
Receipt message: Success
Return message: Success
Return value: [0]
-----
Event logs
Event: {"TransferEvent": [[0, "wheel", "car", 500]], "RepayReceiptEvent": [[0, "25", 500]] }
```

- 交易25的信息更新

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectReceipt 25
-----
Return code: 0
description: transaction executed successfully
Return message: Success
-----
Return values:
[
  0,
  [
    1000,
    0
  ],
  [
    "car",
    "wheel"
  ]
]
```

交易25已还清

- 车企 car 资产额度变为 $9999 + 500 = 10499$ ，轮胎厂 wheel 变为 $2000 - 500 = 1500$

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectAccount car
-----
Return code: 0
description: transaction executed successfully
Return message: Success
-----
Return values:
[
  0,
  10499
]
```

```
[group:1] > call Asset 0xa7820227266c7ea6df8770d56bde68b52d0ca022 selectAccount wheel
-----
Return code: 0
description: transaction executed successfully
Return message: Success
-----
Return values:
[
  0,
  1500
]
```

- 至此功能四 应收账款支付结算上链 验证完毕。

