# 完成的go源码文件

1. cipsMain.go
2. cipsReadLedger.go
3. cipsWriteLedger.go

# 各个go文件中的方法

## cipsMain.go

//定义了合约的数据结构如下：



各变量含义：

--MessageIdentification

交易标识号

InstructingParty

交易发起行

-- DebtorAccount

交易发起人账号

-- DebtorName

交易发起人户名

InstructedParty

交易接收行

-- CreditorAccount

交易接收人账号

-- CreditorName

交易接收人户名

-- Amount

交易金额

-- ContractStatus

交易状态

### main()

// 入口函数

### Init()

//实例化chaincode的时候调用该函数

### Invoke()

// 在cli中使用invoke指令时，调用该函数

### sanitize\_arguments(strs []string) error

//判断输入的字符串字符数是否在0-32个字符之间，不在的话返回错误

## 2.2 cipsReadLedger.go

### 2.2.1 getHistory()

//根据交易标识号，获取该交易的相关所有历史信息

输入：

参数个数：1个

参数类型：string

参数名称：MessageIdentification

输出：

交易的全部历史信息

CLI调用指令例子：

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile $ORDERER\_CA -C mychannel -n mycc -c '{"Args":["getHistory","id0000"]}'

### 2.2.2 getCtrctStateById()

//根据交易号获取账本的当前状态

输入：

参数个数：1个

参数类型：string

参数名称：MessageIdentification

输出：

交易的当前信息

CLI调用指令例子：

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile $ORDERER\_CA -C mychannel -n mycc -c '{"Args":["getCtrctStateById","id0001"]}'

### 2.2.3 getCtrctStateByPingS()

//根据”交易发起行+交易状态”获取满足条件的所有交易信息

输入：

参数个数：2个

参数类型：string

参数名称：InstructingParty，Contractstatus

输出：

满足输入条件的所有交易信息

CLI调用指令例子：

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile $ORDERER\_CA -C mychannel -n mycc -c '{"Args":["getCtrctStateByPingS","A","init"]}'

### 2.2.4 getCtrctStateByPedS()

//根据”交易接收行+交易状态”获取满足条件的所有交易信息

输入：

参数个数：2个

参数类型：string

参数名称：InstructedParty，Contractstatus

输出：

满足输入条件的所有交易信息

CLI调用指令例子：

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile $ORDERER\_CA -C mychannel -n mycc -c '{"Args":["getCtrctStateByPedS","B","init"]}'

### 2.2.5 getCtrctStateByStatus()

//根据”交易状态”获取满足条件的所有交易信息

输入：

参数个数：1个

参数类型：string

参数名称：Contractstatus

输出：

满足输入条件的所有交易信息

CLI调用指令例子：

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile $ORDERER\_CA -C mychannel -n mycc -c '{"Args":["getCtrctStateByStatus","init"]}'

## 2.3 cipsWriteLedger.go

### 2.3.1 Create\_Remittance()

//创建一个交易并存入账本

输入：

参数个数：9个

参数类型：(string, string, string, string, string, string, string, string, string)

参数名称：(MessageIdentification,InstructingParty,DebtorAccount,DebtorName, InstructedParty, CreditorAccount, CreditorName, Amount, Contractstatus )

输出：

创建的交易信息

CLI调用指令例子：

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile $ORDERER\_CA -C mychannel -n mycc -c '{"Args":["Create\_Remittance","id0001","A","q0000","Jack","B","r0001","Rose","10000","init"]}'

### 2.3.2 modifyStatus ()

//更改合约状态（init-confirmed-cleared-remitted-received）

输入：

参数个数：2个

参数类型：(string, string)

参数名称：(MessageIdentification, ContractStatus)

输出：

更改状态后的交易信息

CLI调用指令例子：

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile $ORDERER\_CA -C mychannel -n mycc -c '{"Args":["modifyStatus","id0001","confirm"]}'