供应链金融方案设计及说明

合约中票据的设计如下：

type Bill struct {

    BillInfoID string `json:"BillInfoID"` //票据号码

    BillInfoAmt string `json:"BillInfoAmt"` //票据金额

    BillInfoType string `json:"BillInfoType"` //票据类型

    BillInfoIsseDate string `json:"BillInfoIsseDate"` //票据出票日期

    BillInfoDueDate string `json:"BillInfoDueDate"` //票据到期日期

    DrwrCmID string `json:"DrwrCmID"` //出票人证件号码

    DrwrAcct string `json:"DrwrAcct"` //出票人名称

    AccptrCmID string `json:"AccptrCmID"` //承兑人证件号码

    AccptrAcct string `json:"AccptrAcct"` //承兑人名称

    PyeeCmID string `json:"PyeeCmID"` //收款人证件号码

    PyeeAcct string `json:"PyeeAcct"` //收款人名称

    HodrCmID string `json:"HodrCmID"` //持票人证件号码

    HodrAcct string `json:"HodrAcct"` //持票人名称

    WaitEndorserCmID string `json:"WaitEndorserCmID"` //待背书人证件号码

    WaitEndorserAcct string `json:"WaitEndorserAcct"` //待背书人名称

    RejectEndorserCmID string `json:"RejectEndorserCmID"` //拒绝背书人证件号码

    RejectEndorserAcct string `json:"RejectEndorserAcct"` //拒绝背书人名称

    State string `json:"State"` //票据状态

    History []HistoryItem `json:"History"` //背书历史

}

type HistoryItem struct {

    TxID string `json:"txID"`

    Bill Bill `json:"bill"`

}

票据状态分为

票据新发布、票据等待签收、票据签收成功、票据拒绝签收、票据废弃

合约中对票据的操作有

1 票据发布 issue

接收一个参数 Bill 需要前端传入，具体数值参考Bill数据结构

调用函数后，将Bill的状态改为新发布，后将Bill存入数据库

2 票据背书请求 endorse

接收3个参数

args: 0 – 票据号码; 1 – 待背书人证件号码; 2 – 待背书人名称

调用函数后，添加背书人信息，将Bill状态改为待背书，之后更新数据库中Bill信息

3 背书人接收背书 accept

接收3个参数

args: 0 - 票据号码; 1 – 背书人证件号码; 2 – 背书人名称

调用函数后，将当前持票人信息改为背书人，然后重置Bill中待背书人信息，将Bill状态改为票据签收成功，之后更新数据库中Bill信息

4 背书人拒绝背书

接收3个参数

args: 0 - 票据号码; 1 – 背书人证件号码; 2 – 背书人名称

调用函数后，将Bill中拒绝背书人信息更改为当前背书人，重置Bill中待背书人信息，将Bill状态改为票据拒绝签收，之后更新数据库中Bill信息

5 票据废弃

接收3个参数

args: 0 - 票据号码; 1 – 持票人证件号码; 2 – 持票人名称

调用函数后，将Bill信息重置，并且将Bill状态改为票据废弃，之后更新数据库中Bill信息

6 票据拆分 split

接收参数不定

args: 0 - 票据号码; 1 –拆分后的Bill[1]; 2 – 拆分后的Bill[2]...

函数首先根据args[0]获取到要拆分的票据的信息，在前端选择拆分的个数，之后输入新的Bill的待背书人信息等，同时新的Bill中票据类型设置为2（未拆分票据的默认为1），新拆分的票据号码为就票据加后缀



















