

# 链码调用

业务处理全部保存在 `service` 目录中

```
$ mkdir service
```

定义所需结构体

```
$ vim service/structure.go
```

```
package service

import "github.com/kongyixueyuan.com/bill/blockchain"

type FabricSetupService struct {
    Fabric *blockchain.FabricSetup
}

// 票据数据结构
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			
HoldrCmID	string	`json:"HodrCmID"`	// 当前持
票人证件号码			
HoldrAcct	string	`json:"HodrAcct`	// 当前持
票人名称			
WaitEndorseCmID	string	`json:"WaitEndorseCmID"`	// 待背书
人证件号码			
WaitEndorseAcct	string	`json:"WaitEndorseAcct"`	// 待背书
人名称			
RejectEndorseCmID	string	`json:"RejectEndorseCmID"`	//
拒绝背书人证件号码			

```

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

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

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

// 票据历史信息
type HistoryItem struct {
    TxId    string    `json:"TxId"`
    Bill    Bill      `json:"bill"`
}

```

## 发布票据

```
$ vim service/billService.go
```

```

package service

import (
    "github.com/hyperledger/fabric-sdk-go/api/apitxn/chclient"
    "fmt"
    "encoding/json"
)

// 发布票据
func (setup *FabricSetupService) SaveBill(bill Bill) (string, error)
{
    var args []string
    args = append(args, "issue")

    b, _ := json.Marshal(bill)
}

```

```

// 设置交易请求参数
req := chclient.Request{ChaincodeID: setup.Fabric.ChaincodeID,
Fcn: args[0], Args: [][]byte{b}}

// 执行交易
response, err := setup.Fabric.Client.Execute(req)
if err != nil {
    return "", fmt.Errorf("保存票据时发生错误: %v\n", err)
}

return response.TransactionID.ID, nil
}

```

## 测试发布票据

```
$ vim main.go
```

main.go 完整代码如下:

```

package main

import (
    "os"
    "github.com/kongyixueyuan.com/bill/blockchain"
    "fmt"
    "github.com/kongyixueyuan.com/bill/service"
)

func main() {
    fSetup := blockchain.FabricSetup{
        OrgAdmin:      "Admin",
        OrgName:       "Org1",
        ConfigFile:    "config.yaml",

        ChannelID:     "mychannel",
        ChannelConfig: os.Getenv("GOPATH") +
"/src/github.com/kongyixueyuan.com/bill/fixtures/artifacts/channel.t
x",
    }
}

```

```

// 链码相关
ChaincodeID:      "billcc",
ChaincodeGoPath:   os.Getenv("GOPATH"),
ChaincodePath:
"github.com/kongyixueyuan.com/bill/chaincode/",

    UserName:      "User1",
}

err := fSetup.Initialize()
if err != nil {
    fmt.Printf("无法初始化SDK: %v\n", err)
}

err = fSetup.InstallAndInstantiateCC()
if err != nil {
    fmt.Printf("无法安装及实例化链码: %v\n", err)
}

// =====测试开始
=====
// 发布票据
bill := service.Bill{
    BillInfoID:      "BOC10000001",
    BillInfoAmt:      "222",
    BillInfoType:      "111",
    BillInfoIsseDate:  "20180501",
    BillInfoDueDate:   "20180503",
    DrwrCmID:          "111",
    DrwrAcct:           "111",
    AccptrCmID:         "111",
    AccptrAcct:         "111",
    PyeeCmID:           "111",
    PyeeAcct:           "111",
    HoldrCmID:          "BCMID",
    HoldrAcct:          "B公司",
}

```

```

fsService := new(service.FabricSetupService)
fsService.Fabric = &fSetup

// 发布票据
resp, err := fsService.SaveBill(bill)
if err != nil {
    fmt.Printf("发布票据失败: %v\n", err)
}else {
    fmt.Println("发布票据成功 =====> " + resp)
}
}

```

## 测试发布

在命令行中执行 `make`

```

Build ...
Build done
Start environment ...
Creating network "fixtures_default" with the default driver
Creating ca.org1.kongyixueyuan.com
Creating orderer.kongyixueyuan.com
Creating peer1.org1.kongyixueyuan.com
Creating peer0.org1.kongyixueyuan.com
Sleep 15 seconds in order to let the environment setup correctly
Environment up
Start app ...
开始初始化SDK.....
[fabric_sdk_go] 2018/06/05 09:15:10 UTC - config.initConfig -> INFO config fabric_sdk_go logging level is set to: INFO
SDK初始化成功
开始安装链码.....
链码安装成功!
开始实例化链码.....
链码实例化成功!
[fabric_sdk_go] 2018/06/05 09:15:54 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
[fabric_sdk_go] 2018/06/05 09:15:54 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
发布票据成功 =====> 91972ccdab6a8b8ec7daae91d67fa29d55637dd5a87cdd0bee338141c2a132d8
kevin@kevin-hf:~/go/src/github.com/kongyixueyuan.com/bill$

```

## 查询持票人的票据

就像一个数据库一样，链码被插入并可以交互。开始测试票据查询。

把所有的票据操作函数放在位于 `service` 文件夹中一个名为 `billService.go` 的文件中：

```
$ vim service/billService.go
```

```
[...]
```

```
// 根据当前持票人证件号码，批量查询票据
func (setup *FabricSetupService) FindBills(holderCmId string)
([]byte, error) {

    var args []string
    // 1. 将所需参数添加至args中

    // 2. 设置查询的请求参数

    // 3. 执行查询

    // 4. 返回
}
```

添加下面代码到 `main.go` 测试新功能：

```
$ vim main.go
```

在 `main.go` 的 `main` 方法中添加如下内容

```
import(
    [...]
    "encoding/json"
)

func main() {

    [...]

    // 根据当前用户的证件号码查询票据列表
    result, err := fsService.FindBills("BCMID")

    // 查看返回结果
}
```

再运行一下：

```
$ make
```

```
[fabric_sdk_go] 2018/06/05 09:29:57 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
[fabric_sdk_go] 2018/06/05 09:29:57 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
发布票据成功 =====> 328c40873658cfe6715866aa5b3e2b7693ed0b3110f8fdd4d55a866ce2cad786
查询票据信息.....
[fabric_sdk_go] 2018/06/05 09:29:59 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
[fabric_sdk_go] 2018/06/05 09:29:59 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
票据信息查询完毕!
根据当前用户的证件号码查询票据列表成功
=====> [{"BillInfoID":"POC10000500","BillInfoAmt":"222","BillInfoType":"111","BillInfoIsseDate":"20180501","billInfoDueDate":"20180503","DrwrCmID":"111","DrwrCmAcct":"111","AccptrCmID":"111","AccptrAcct":"111","PyeeCmID":"111","PyeeAcct":"111","HodrCmID":"BCMID","HodrAcct":"B公司","WaitEndorserCmID":"","WaitEndorserAcct":"","RejectEndorserCmID":"","RejectEndorserAcct":"","State":"NewPublish","History":null}]
kevin@kevin-hf:~/go/src/github.com/kongyixueyuan.com/bill$
```

## 票据背书

与票据相关的业务处理方法全部定义在 `endorseService.go` 文件中

添加票据背书的实现代码

```
$ vim service/endorseService.go
```

```
package service

import (
    "github.com/hyperledger/fabric-sdk-go/api/apitxn/chclient"
    "fmt"
)

// 票据背书请求
// args: 0 - Bill_No; 1 - endorserCmId(待背书人ID); 2 -
// endorserAcct(待背书人名称)
```



```
func (setup *FabricSetupService) Endorse(bill_no string, endorseCmId
string, endorseAcct string) (string, error) {
    var args []string
    // 1. 将所需参数添加至args中

    // 2. 设置查询的请求参数

    // 3. 执行查询

    // 4. 返回
}
```

添加下面代码到 `main.go` 测试新功能：

```
$ vim main.go
```

在 `main.go` 的 `main` 方法中添加如下内容

```
func main() {

[...]
```

    // 票据背书请求

```
    resp, err = fsService.Endorse("BOC10000001", "CCMID", "C公司")
    // 查看返回结果
}
```

再运行一下：

```
$ make
```

```
发布票据成功 =====> 404bea1b1c7910460fa8f029da90a4b2dff807965f492a1e8321c
2c4ffc4de49
[fabric_sdk_go] 2018/06/05 13:12:22 UTC - msp.(*bccspmsp).DeserializeIdentit
y -> INFO Obtaining identity
[fabric_sdk_go] 2018/06/05 13:12:22 UTC - msp.(*bccspmsp).DeserializeIdentit
y -> INFO Obtaining identity
根据当前用户的证件号码查询票据列表成功
=====> [{"BillInfoID":"POC10000500","BillInfoAmt":"222","BillInfoType":"1
11","BillInfoIsseDate":"20180501","billInfoDueDate":"20180503","DrwrCmID":"11
1","DrwrCmAcct":"111","AccptrCmID":"111","AccptrAcct":"111","PyeeCmID":"111",
"PyeeAcct":"111","HodrCmID":"BCMID","HodrAcct":"B公司","WaitEndorserCmID":"","
"WaitEndorserAcct":"","RejectEndorserCmID":"","RejectEndorserAcct":"","State"
:"NewPublish","History":null}]
[fabric_sdk_go] 2018/06/05 13:12:22 UTC - msp.(*bccspmsp).DeserializeIdentit
y -> INFO Obtaining identity
[fabric_sdk_go] 2018/06/05 13:12:22 UTC - msp.(*bccspmsp).DeserializeIdentit
y -> INFO Obtaining identity
票据背书请求成功
=====> {"Code":0,"Dec":"背书请求成功"}
kevin@kevin-hf:~/go/src/github.com/kongyixueyuan.com/bill$
```

## 查询待背书票据

在 `service/endorseService.go` 文件中添加查询待背书票据实现方法

```
$ vim service/endorseService.go
```

```
[...]
```

```
// 根据待背书人证件号码，查询当前用户的待背书票据
```

```
func (setup *FabricSetupService) FindWaitBills(endorserCmId string)
```

```
([]byte, error){
```

```
    var args []string
```

```
    // 1. 将所需参数添加至args中
```

```
    // 2. 设置查询的请求参数
```

```
    // 3. 执行查询
```

```
    // 4. 返回
```

```
}
```

添加下面代码到 `main.go` 测试新功能：

```
$ vim main.go
```

在 `main.go` 的 `main` 方法中添加如下内容

```
func main() {  
  
    [...]  
  
    // 根据待背书人证件号码，查询当前用户的待背书票据  
    result, err = fsService.FindWaitBills("CCMID")  
    // 查看返回结果  
}
```

再运行一下：

```
$ make
```

```
"WaitEndorserAcct":"","RejectEndorserCmID":"","RejectEndorserAcct":"","State"  
:"NewPublish","History":null}}  
[fabric_sdk_go] 2018/06/05 13:23:24 UTC - msp.(*bccspmsp).DeserializeIdentit  
y -> INFO Obtaining identity  
[fabric_sdk_go] 2018/06/05 13:23:24 UTC - msp.(*bccspmsp).DeserializeIdentit  
y -> INFO Obtaining identity  
票据背书请求成功  
=====> {"Code":0,"Dec":"背书请求成功"}  
[fabric_sdk_go] 2018/06/05 13:23:27 UTC - msp.(*bccspmsp).DeserializeIdentit  
y -> INFO Obtaining identity  
[fabric_sdk_go] 2018/06/05 13:23:27 UTC - msp.(*bccspmsp).DeserializeIdentit  
y -> INFO Obtaining identity  
查询当前用户的待背书票据成功  
=====> [{"BillInfoID":"POC10000500","BillInfoAmt":"222","BillInfoType":"1  
11","BillInfoIsseDate":"20180501","billInfoDueDate":"20180503","DrwrCmID":"11  
1","DrwrCmAcct":"111","AccptrCmID":"111","AccptrAcct":"111","PyeeCmID":"111",  
"PyeeAcct":"111","HodrCmID":"BCMID","HodrAcct":"B公司","WaitEndorserCmID":"CC  
MID","WaitEndorserAcct":"C公司","RejectEndorserCmID":"","RejectEndorserAcct":  
"", "State":"EndrWaitSign","History":null}}  
kevin@kevin-hf:~/go/src/github.com/kongyixueyuan.com/bill$ |
```

## 票据背书签收

在 `service/endorseService.go` 文件中添加票据背书签收实现方法

```
$ vim service/endorseService.go
```

```
[...]
```

```
// 票据背书签收
// args: 0 - Bill_No; 1 - endorserCmId(待背书人ID); 2 -
endorserAcct(待背书人名称)
func (setup *FabricSetupService) EndorseAccept(bill_no string,
endorserCmId string, endorserAcct string) (string, error){
    var args []string
    // 1. 将所需参数添加至args中

    // 2. 设置查询的请求参数

    // 3. 执行查询

    // 4. 返回
}
```

添加下面代码到 `main.go` 测试新功能：

```
$ vim main.go
```

在 `main.go` 的 main 方法中添加如下内容

```
func main() {

[...]
```

```
    // 票据背书签收
    resp, err = fsService.EndorseAccept("BOC10000001", "CCMID", "C公
司")
    // 查看返回结果
}
```

再运行一下：

```
$ make
```

票据背书请求成功

```
=====> {"Code":0,"Dec":"背书请求成功"}
[fabric_sdk_go] 2018/06/05 13:35:13 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
[fabric_sdk_go] 2018/06/05 13:35:13 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
查询当前用户的待背书票据成功
=====> [{"BillInfoID":"POC10000500","BillInfoAmt":"222","BillInfoType":"111","BillInfoIsseDate":"20180501","billInfoDueDate":"20180503","DrwrCmID":"111","DrwrCmAcct":"111","AccptrCmID":"111","AccptrAcct":"111","PyeeCmID":"111","PyeeAcct":"111","HodrCmID":"BCMID","HodrAcct":"B公司","WaitEndorserCmID":"CCMID","WaitEndorserAcct":"C公司","RejectEndorserCmID":"","RejectEndorserAcct":"","State":"EndrWaitSign","History":null}]
[fabric_sdk_go] 2018/06/05 13:35:13 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
[fabric_sdk_go] 2018/06/05 13:35:13 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
票据背书签收成功
=====> {"Code":0,"Dec":"票据背书签收成功"}
kevin@kevin-hf:~/go/src/github.com/kongyixueyuan.com/bill$
```

## 票据背书拒收

在 `service/endorseService.go` 文件中添加票据背书签收实现方法

```
$ vim service/endorseService.go
```

```
[...]

// 票据背书拒签(拒绝背书)
// args: 0 - bill_NO; 1 - endorserCmId(待背书人ID); 2 -
endorserAcct(待背书人名称)
func (setup *FabricSetupService) EndorseReject(bill_no string,
endorseCmId string, endorseAcct string) (string, error) {
    var args []string
    // 1. 将所需参数添加至args中

    // 2. 设置查询的请求参数

    // 3. 执行查询

    // 4. 返回
}
```

下面代码到 `main.go` 测试新功能：

```
$ vim main.go
```

在 `main.go` 的 `main` 方法中添加如下内容

```
func main() {

[...]

    // 票据背书拒签
    resp, err = fsService.EndorseReject("BOC10000001", "CCMID", "C公
司")
    // 查看返回结果
}
```

再运行一下：

```
$ make
```

查询当前用户的待背书票据成功

```
=====> [{"BillInfoID":"POC10000500","BillInfoAmt":"222","BillInfoType":"111","BillInfoIsseDate":"20180501","billInfoDueDate":"20180503","DrwrCmID":"111","DrwrCmAcct":"111","AcceptorCmID":"111","AcceptorAcct":"111","PyeeCmID":"111","PyeeAcct":"111","HodrCmID":"BCMID","HodrAcct":"B公司","WaitEndorserCmID":"CCMID","WaitEndorserAcct":"C公司","RejectEndorserCmID":"","RejectEndorserAcct":"","State":"EndrWaitSign","History":null}]
```

```
[fabric_sdk_go] 2018/06/05 13:48:13 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
```

```
[fabric_sdk_go] 2018/06/05 13:48:13 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
```

票据背书签收成功

```
=====> {"Code":0,"Dec":"票据背书签收成功"}
```

```
[fabric_sdk_go] 2018/06/05 13:48:15 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
```

```
[fabric_sdk_go] 2018/06/05 13:48:15 UTC - msp.(*bccspmsp).DeserializeIdentity -> INFO Obtaining identity
```

票据背书拒签成功

```
=====> {"Code":0,"Dec":"票据背书拒签成功"}
```

```
kevin@kevin-hf:~/go/src/github.com/kongyixueyuan.com/bill$ |
```

## 根据票据号码查询票据信息

在 `service/billService.go` 文件中添加票据背书签收实现方法

```
$ vim service/billService.go
```

```
[...]
```

```
// 根据票据号码获取票据状态及该票据的背书历史
```

```
func (setup *FabricSetupService) FindBillByNo(bill_no string)
([]byte, error) {
    var args []string
    // 1. 将所需参数添加至args中

    // 2. 设置查询的请求参数

    // 3. 执行查询

    // 4. 返回
}
```

添加下面代码到 `main.go` 测试新功能：

```
$ vim main.go
```

在 `main.go` 的 `main` 方法中添加如下内容

```
func main() {

[...]
```

```
    // 根据票据号码获取票据状态及该票据的背书历史
    result, err = fsService.FindBillByNo("BOC10000001")
    // 查看返回结果
}
```

再运行一下：

```
$ make
```



```
y -> INFO Obtaining identity
根据票据号码获取票据状态及该票据的背书历史查询成功
=====> {"BillInfoID":"POC10000500","BillInfoAmt":"222","BillInfoType":"111",
"BillInfoIsseDate":"20180501","billInfoDueDate":"20180503","DrwrCmID":"111",
"DrwrCmAcct":"111","AccptrCmID":"111","AccptrAcct":"111","PyeeCmID":"111",
"PyeeAcct":"111","HodrCmID":"CCMID","HodrAcct":"C公司","WaitEndorserCmID":"","
WaitEndorserAcct":"","RejectEndorserCmID":"CCMID","RejectEndorserAcct":"C公司",
"State":"EndrReject","History":[{"TxId":"f5bcee0f7bb71d8262a37244474f67cf19
895d8ec2ef746a492a31dfae4dbe82","bill":{"BillInfoID":"POC10000500","BillInfoA
mt":"222","BillInfoType":"111","BillInfoIsseDate":"20180501","billInfoDueDate
":"20180503","DrwrCmID":"111","DrwrCmAcct":"111","AccptrCmID":"111","AccptrAc
ct":"111","PyeeCmID":"111","PyeeAcct":"111","HodrCmID":"BCMID","HodrAcct":"B
公司","WaitEndorserCmID":"","WaitEndorserAcct":"","RejectEndorserCmID":"","Re
jectEndorserAcct":"","State":"NewPublish","History":null}},{TxId":"00cbdd757
e164099d5175d6d5e7b84d0adc2badf9a94a4e11807c41d324f0e81","bill":{"BillInfoID"
:"POC10000500","BillInfoAmt":"222","BillInfoType":"111","BillInfoIsseDate":"2
0180501","billInfoDueDate":"20180503","DrwrCmID":"111","DrwrCmAcct":"111","Ac
cptrCmID":"111","AccptrAcct":"111","PyeeCmID":"111","PyeeAcct":"111","HodrCmI
D":"BCMID","HodrAcct":"B公司","WaitEndorserCmID":"CCMID","WaitEndorserAcct":"
C公司","RejectEndorserCmID":"","RejectEndorserAcct":"","State":"EndrWaitSign"
,"History":null}},{TxId":"c1a1021a2b437ea25b015c7cffffc0ecc15d03e6652471e9a9c
63cce4cle6419e","bill":{"BillInfoID":"POC10000500","BillInfoAmt":"222","BillI
nfoType":"111","BillInfoIsseDate":"20180501","billInfoDueDate":"20180503","Dr
wrCmID":"111","DrwrCmAcct":"111","AccptrCmID":"111","AccptrAcct":"111","PyeeC
mID":"111","PyeeAcct":"111","HodrCmID":"CCMID","HodrAcct":"C公司","WaitEndors
erCmID":"","WaitEndorserAcct":"","RejectEndorserCmID":"","RejectEndorserAcct"
:"","State":"EndrSigned","History":null}},{TxId":"d37388d5729733ecf19ae387d2
2e8e6bb4012593751926c6bee15d858582c8ea","bill":{"BillInfoID":"POC10000500","B
illInfoAmt":"222","BillInfoType":"111","BillInfoIsseDate":"20180501","billInf
oDueDate":"20180503","DrwrCmID":"111","DrwrCmAcct":"111","AccptrCmID":"111",
"AccptrAcct":"111","PyeeCmID":"111","PyeeAcct":"111","HodrCmID":"CCMID","HodrA
cct":"C公司","WaitEndorserCmID":"","WaitEndorserAcct":"","RejectEndorserCmID"
:"CCMID","RejectEndorserAcct":"C公司","State":"EndrReject","History":null}}}]
kevin@kevin-hf:~/go/src/github.com/kongyixueyuan.com/bill$ |
```



区块链部落

专注于区块链技术



识别图中二维码关注我们