

链码开发

安装及实例化链码

为了安装和实例化chaincode，需要在应用程序中添加一些代码。编辑 `blockchain/startInit.go` 为如下代码：

```
$ vim blockchain/startInit.go
```

startInit.go添加内容如下:

```
package blockchain

import (
    "github.com/hyperledger/fabric-sdk-go/api/apitxn/resmgmtclient"
    "github.com/hyperledger/fabric-sdk-go/pkg/fabsdk"
    "fmt"
    "github.com/hyperledger/fabric-sdk-go/pkg/config"
    "github.com/hyperledger/fabric-sdk-go/api/apitxn/chmgmtclient"
    "time"
    "github.com/hyperledger/fabric-sdk-go/pkg/fabric-
client/ccpackager/gopackager"
    "github.com/hyperledger/fabric-sdk-
go/third_party/github.com/hyperledger/fabric/common/cauthdsl"
    "github.com/hyperledger/fabric-sdk-go/api/apitxn/chclient"
```

```

)

const chaincodeVersion = "1.0"

type FabricSetup struct {
    [.....]

    // 链码相关
    ChaincodeID      string
    ChaincodeGoPath  string
    ChaincodePath     string
    UserName          string
    Client            chclient.ChannelClient
}

[.....]

// 安装并实例化链码
func (setup *FabricSetup) InstallAndInstantiateCC() error {
    fmt.Println("开始安装链码.....")

    // 创建一个新的链码包并使用我们的链码初始化
    ccPkg, err := gopackager.NewCCPackage(setup.ChaincodePath,
    setup.ChaincodeGoPath)
    if err != nil {
        return fmt.Errorf("创建链码包失败: %v\n", err)
    }

    // 指定要安装链码的各项参数
    installCCReq := resmgmtclient.InstallCCRequest{Name:
    setup.ChaincodeID, Path: setup.ChaincodePath, Version:
    chaincodeVersion, Package: ccPkg}

    // 在Org Peer上安装链码
    _, err = setup.admin.InstallCC(installCCReq)
    if err != nil {
        return fmt.Errorf("安装链码失败: %v\n", err)
    }
}

```

```

fmt.Println("链码安装成功!")
fmt.Println("开始实例化链码.....")

// 设置链码策略
ccPolicy := cauthdsl.SignedByAnyMember([]string{"Org1MSP"})

// 指定实例化链码相关参数
instantiateCCReq := resmgmtclient.InstantiateCCRequest{Name:
setup.ChaincodeID, Path: setup.ChaincodePath, Version:
chaincodeVersion, Args: [][]byte{[]byte("init")}, Policy: ccPolicy}
// 实例化链码
err = setup.admin.InstantiateCC(setup.ChannelID,
instantiateCCReq)
if err != nil {
    return fmt.Errorf("实例化链码失败: %v\n", err)
}

// 创建通道客户端用于查询与执行事务
setup.Client, err =
setup.sdk.NewClient(fabsdk.WithUser(setup.UserName)).Channel(setup.C
hannelID)
if err != nil {
    return fmt.Errorf("创建新的通道客户端失败: %v\n", err)
}

fmt.Println("链码实例化成功!")
return nil
}

```

提示： 请注意链码版本，如果想更新链码，请文件中设置的版本号。否则，网络将保持相同的链码。

现在需要修改 main.go 文件来调用新函数

```
$ cd $GOPATH/src/github.com/kongyixueyuan.com/bill
$ vim main.go
```

```
package main

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

func main() {
    fSetup := blockchain.FabricSetup{

        // 公共参数
        OrgName:      "Org1",
        OrgAdmin:     "Admin",
        ConfigFile:   "config.yaml",

        // 通道相关参数
        ChannelID:     "mychannel",
        ChannelConfig: os.Getenv("GOPATH") +
"/src/github.com/kongyixueyuan.com/bill/fixtures/artifacts/channel.tx",

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

        // 指定用户
        UserName:     "User1",

    }
}
```

```
err := fSetup.Initialize()
if err != nil {
    fmt.Printf("无法初始化Fabric 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)
}

}
```

使用 `make` 命令来测试：

```
$ make
```

Chaincode安装并实例化

出现错误:

```
Unable to install and instantiate the chaincode:failed to
instantiate the chaincode: send instantiate chaincode proposal
failed: Multiple errors occurred:Transaction processor
(localhost:7051) returned error for txID
'a1aaf019ed7840faeafbf88d4b802509db54694d2836a372b7934bdf7412abed':g
RPC Transport Status Code: (2) Unknown. Description:timeout expired
while starting chaincode
```

```
fabgoproject:1.0(networkid:fabricpro,peerid:peer0.org1.example.com,t
x:a1aaf019ed7840faeafbf88d4b802509db54694d2836a372b7934bdf7412abed)T
ransaction processor (localhost:8051) returned error for txID
'a1aaf019ed7840faeafbf88d4b802509db54694d2836a372b7934bdf7412abed':g
RPC Transport Status Code: (2) Unknown. Description:timeout expired
while starting
chaincodefabgoproject:1.0(networkid:fabricpro,peerid:peer1.org1.exam
ple.com,tx:a1aaf019ed7840faeafbf88d4b802509db54694d2836a372b7934bdf7
412abed)
```

需要将fixtures/docker-compose.yaml文件进行修改
CORE_PEER_ADDRESSAUTODETECT 必须设置为 true

若出现错误

```
: exit status 128
    failed to set up sources from the following URLs:
http://github.com/hyperledger/fabric-sdk-go
: remote repository at http://github.com/hyperledger/fabric-sdk-go
does not exist, or is inaccessible: fatal: unable to access
'http://github.com/hyperledger/fabric-sdk-go/': Could not resolve
host: github.com
: exit status 128
Makefile:10: recipe for target 'build' failed
make: *** [build] Error 1
```

这是网络差的问题

```
Build ...
Solving failure: No versions of github.com/google/certificate-
transparency-go met constraints:
    5ab67e519c93568ac3ee50fd6772a5bcf8aa460d: unable to deduce
repository and source type for "golang.org/x/crypto": unable to read
metadata: unable to fetch raw metadata: failed HTTP request to URL
"http://golang.org/x/crypto?go-get=1": Get
http://golang.org/x/crypto?go-get=1: dial tcp 216.239.37.1:80: i/o
timeout
    master: Could not introduce github.com/google/certificate-
transparency-go@master, as it is not allowed by constraint
5ab67e519c93568ac3ee50fd6772a5bcf8aa460d from project
github.com/cloudflare/cfssl.
Makefile:10: recipe for target 'build' failed
make: *** [build] Error 1
```



区块链部落

专注于区块链技术



识别图中二维码关注我们