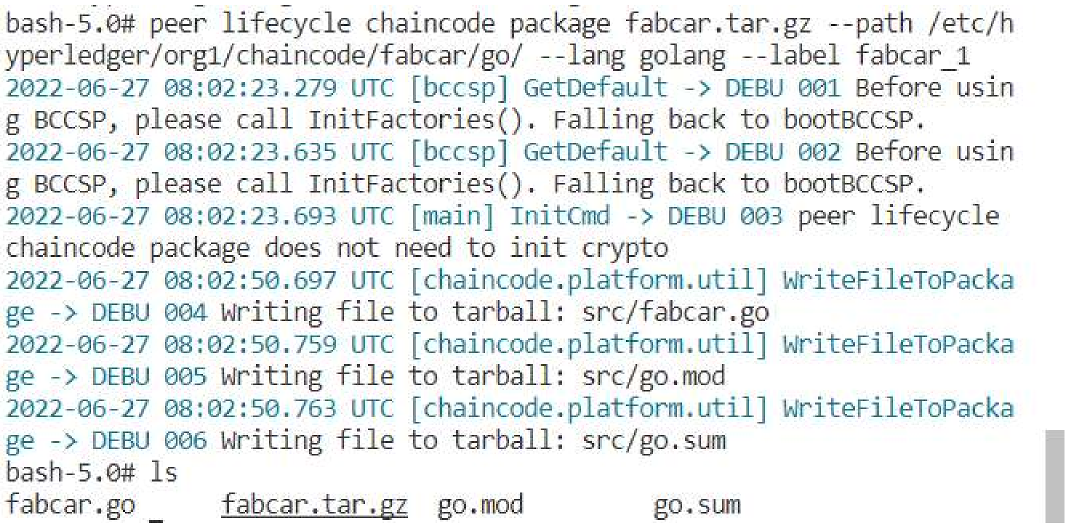
区块链 实验4 报告

PB20000096 潘廷岳

> c档

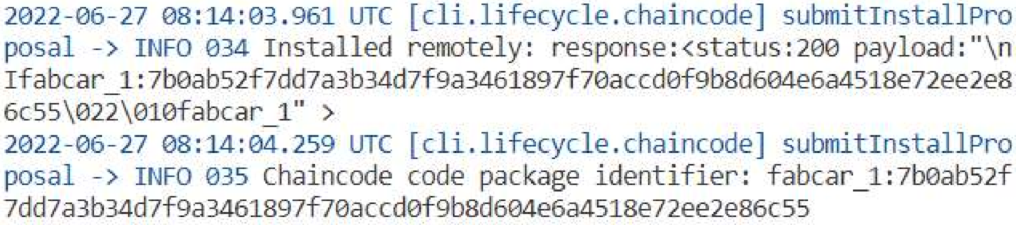
- 链码打包

```
peer lifecycle chaincode package fabcar.tar.gz --path /etc/hyperledger/org1/chaincode/fabcar/go/ --lang golang --label fabcar_1
```



```
bash-5.0# peer lifecycle chaincode package fabcar.tar.gz --path /etc/h
yperledger/org1/chaincode/fabcar/go/ --lang golang --label fabcar_1
2022-06-27 08:02:23.279 UTC [bccsp] GetDefault -> DEBU 001 Before usin
g BCCSP, please call InitFactories(). Falling back to bootBCCSP.
2022-06-27 08:02:23.635 UTC [bccsp] GetDefault -> DEBU 002 Before usin
g BCCSP, please call InitFactories(). Falling back to bootBCCSP.
2022-06-27 08:02:23.693 UTC [main] InitCmd -> DEBU 003 peer lifecycle
chaincode package does not need to init crypto
2022-06-27 08:02:50.697 UTC [chaincode.platform.util] WriteFileToPacka
ge -> DEBU 004 Writing file to tarball: src/fabcar.go
2022-06-27 08:02:50.759 UTC [chaincode.platform.util] WriteFileToPacka
ge -> DEBU 005 Writing file to tarball: src/go.mod
2022-06-27 08:02:50.763 UTC [chaincode.platform.util] WriteFileToPacka
ge -> DEBU 006 Writing file to tarball: src/go.sum
bash-5.0# ls
fabcar.go      fabcar.tar.gz  go.mod         go.sum
```

```
//使用admin的证书来进行链码安装操作
export CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/org1/admin/msp
//安装链码
peer lifecycle chaincode install fabcar.tar.gz
//查询安装的链码
peer lifecycle chaincode queryinstalled
```

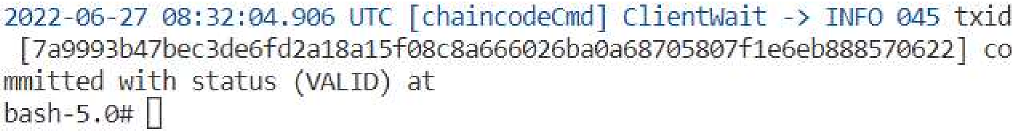


```
2022-06-27 08:14:03.961 UTC [cli.lifecycle.chaincode] submitInstallPro
posal -> INFO 034 Installed remotely: response:<status:200 payload:"\n
Ifabcar_1:7b0ab52f7dd7a3b34d7f9a3461897f70accd0f9b8d604e6a4518e72ee2e8
6c55\022\010fabcar_1" >
2022-06-27 08:14:04.259 UTC [cli.lifecycle.chaincode] submitInstallPro
posal -> INFO 035 Chaincode code package identifier: fabcar_1:7b0ab52f
7dd7a3b34d7f9a3461897f70accd0f9b8d604e6a4518e72ee2e86c55
```

```
2022-06-27 08:16:26.538 UTC [msp.identity] Sign -> DEBU 033 Sign: digest: C44323011E17EB07BCA6B6BC341C21B2B83F88AE8EACFCEF345DC8DA411529DC
Installed chaincodes on peer:
Package ID: fabcar_1:7b0ab52f7dd7a3b34d7f9a3461897f70accd0f9b8d604e6a4518e72ee2e86c55, Label: fabcar_1
```

- 链码安装

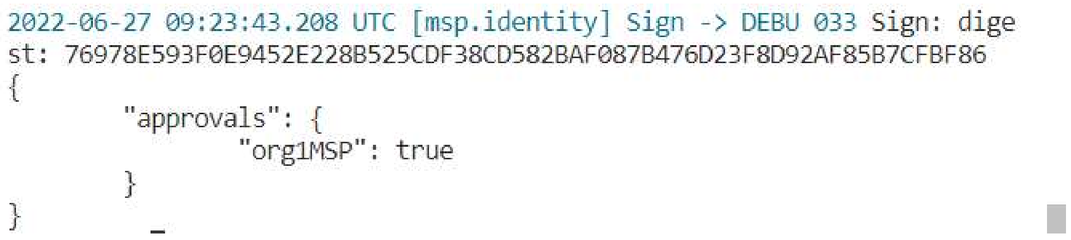
```
export CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/org1/admin/msp

peer lifecycle chaincode queryinstalled

export VERSION=1

export PACKAGE_ID=fabcar_1:7b0ab52f7dd7a3b34d7f9a3461897f70accd0f9b8d604e6a4518e72ee2e86c55

// tls证书
export ORDERER_CA=/etc/hyperledger/org1/peer2/tls-msp/tlscacerts/tls-172-16-4-35-7052.pem
export CHANNEL_NAME=mychannel

peer lifecycle chaincode approveformyorg -o orderer1-org0:7050 --ordererTLSHostnameOverride orderer1-org0 --tls --cafile /etc/hyperledger/org1/peer2/tls-msp/tlscacerts/tls-172-16-4-35-7052.pem --ch
```



```
2022-06-27 08:32:04.906 UTC [chaincodeCmd] ClientWait -> INFO 045 txid
 [7a9993b47bec3de6fd2a18a15f08c8a666026ba0a68705807f1e6eb888570622] co
mmitted with status (VALID) at
bash-5.0# 
```

peer lifecycle chaincode checkcommitreadiness --channelID mychannel --name fabcar --version 1 --sequence 66 --output json

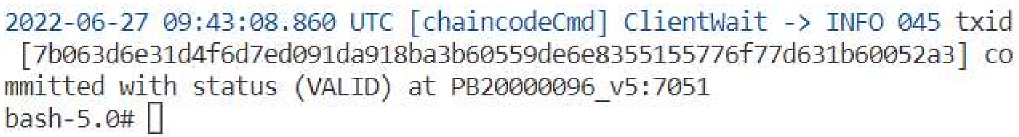
- 查看准入信息



```
2022-06-27 09:23:43.208 UTC [msp.identity] Sign -> DEBU 033 Sign: dige
st: 76978E593F0E9452E228B525CDF38CD582BAF087B476D23F8D92AF85B7CFBF86
{
        "approvals": {
                "org1MSP": true
        }
}
```

```
peer lifecycle chaincode commit -o orderer1-org0:7050 --ordererTLSHostnameOverride orderer1-org0 --tls --cafile /etc/hyperledger/org1/peer2/tls-msp/tlscacerts/tls-172-16-4-35-7052.pem --channelID m
```

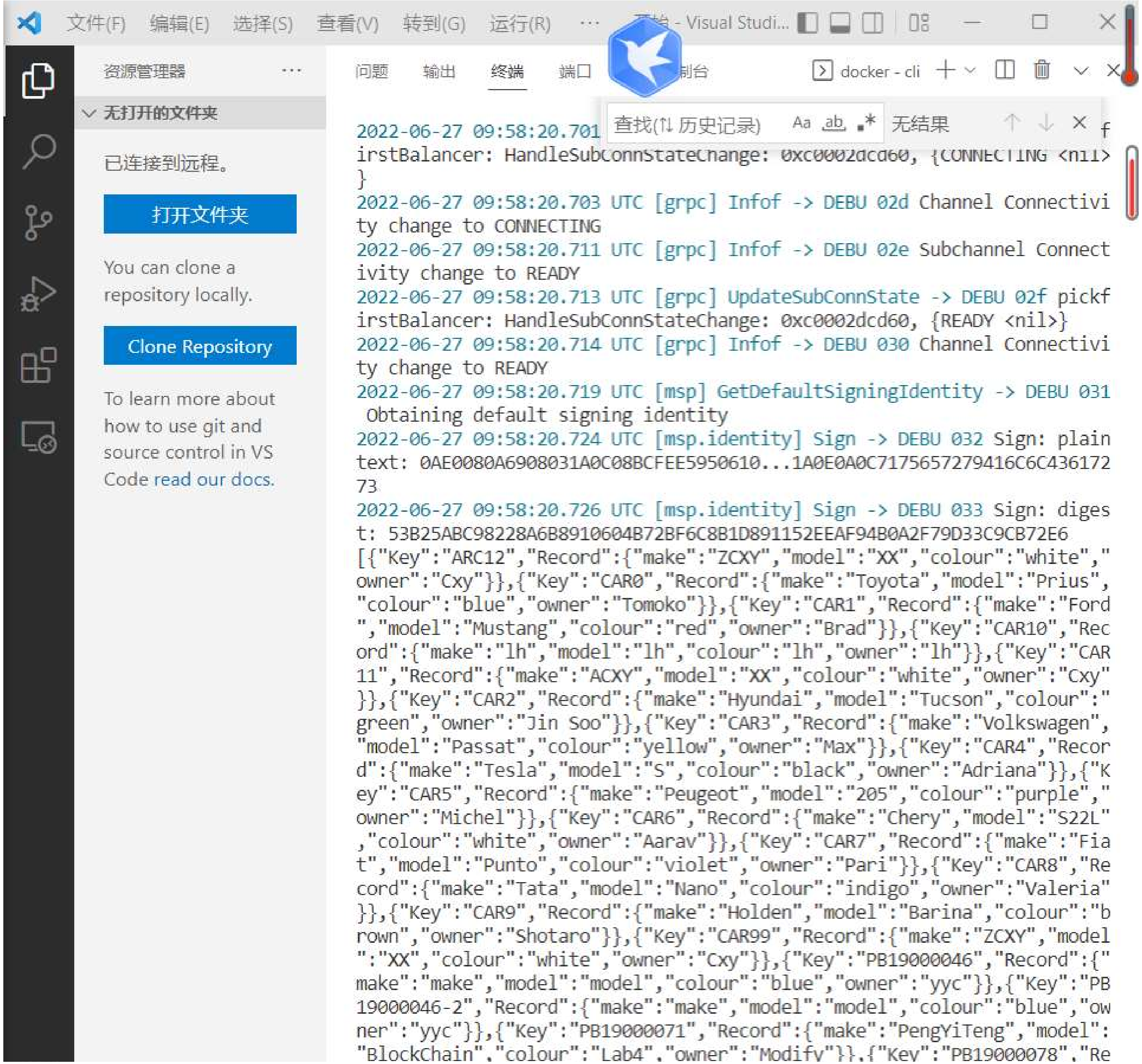
- 成功上链



```
2022-06-27 09:43:08.860 UTC [chaincodeCmd] ClientWait -> INFO 045 txid
 [7b063d6e31d4f6d7ed091da918ba3b60559de6e8355155776f77d631b60052a3] co
mmitted with status (VALID) at PB20000096_v5:7051
bash-5.0# 
```

```
// 首先需要初始化账本 peerAddresses对应当前节点
peer chaincode invoke -o orderer1-org0:7050 --ordererTLSHostnameOverride orderer1-org0 --tls --cafile /etc/hyperledger/org1/peer2/tls-msp/tlscacerts/tls-172-16-4-35-7052.pem -C mychannel -n fabcar
//查询资产
peer chaincode query -C mychannel -n fabcar_v1 -c '{"Args":["queryAllCars"]}'
```

- 查询结果

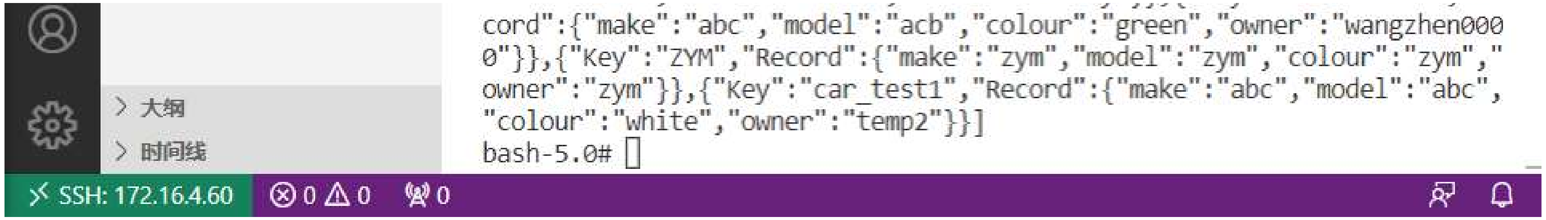


已连接到远程。

You can clone a repository locally.

To learn more about how to use git and source control in VS Code read our docs.

```
2022-06-27 09:58:20.701                                                 f
irstBalancer: HandleSubConnStateChange: 0xc0002dcd60, {CONNECTING <nil>
}
2022-06-27 09:58:20.703 UTC [grpc] Infof -> DEBU 02d Channel Connectivi
ty change to CONNECTING
2022-06-27 09:58:20.711 UTC [grpc] Infof -> DEBU 02e Subchannel Connect
ivity change to READY
2022-06-27 09:58:20.713 UTC [grpc] UpdateSubConnState -> DEBU 02f pickf
irstBalancer: HandleSubConnStateChange: 0xc0002dcd60, {READY <nil>}
2022-06-27 09:58:20.714 UTC [grpc] Infof -> DEBU 030 Channel Connectivi
ty change to READY
2022-06-27 09:58:20.719 UTC [msp] GetDefaultSigningIdentity -> DEBU 031
 Obtaining default signing identity
2022-06-27 09:58:20.724 UTC [msp.identity] Sign -> DEBU 032 Sign: plain
text: 0AE0080A6908031A0C08BCFEE5950610...1A0E0A0C7175657279416C6C436172
73
2022-06-27 09:58:20.726 UTC [msp.identity] Sign -> DEBU 033 Sign: diges
t: 53B25ABC98228A6B8910604B72BF6C8B1D891152EEAF94B0A2F79D33C9CB72E6
[{"Key":"ARC12","Record":{"make":"ZCXY","model":"XX","colour":"white","
owner":"Cxy"}},{"Key":"CAR0","Record":{"make":"Toyota","model":"Prius",
"colour":"blue","owner":"Tomoko"}},{"Key":"CAR1","Record":{"make":"Ford
","model":"Mustang","colour":"red","owner":"Brad"}},{"Key":"CAR10","Rec
ord":{"make":"lh","model":"lh","colour":"lh","owner":"lh"}},{"Key":"CAR
11","Record":{"make":"ACXY","model":"XX","colour":"white","owner":"Cxy"
}},{"Key":"CAR2","Record":{"make":"Hyundai","model":"Tucson","colour":"
green","owner":"Jin Soo"}},{"Key":"CAR3","Record":{"make":"Volkswagen",
"model":"Passat","colour":"yellow","owner":"Max"}},{"Key":"CAR4","Recor
d":{"make":"Tesla","model":"S","colour":"black","owner":"Adriana"}},{"K
ey":"CAR5","Record":{"make":"Peugeot","model":"205","colour":"purple","
owner":"Michel"}},{"Key":"CAR6","Record":{"make":"Chery","model":"S22L"
,"colour":"white","owner":"Aarav"}},{"Key":"CAR7","Record":{"make":"Fia
t","model":"Punto","colour":"violet","owner":"Pari"}},{"Key":"CAR8","Re
cord":{"make":"Tata","model":"Nano","colour":"indigo","owner":"Valeria"
}},{"Key":"CAR9","Record":{"make":"Holden","model":"Barina","colour":"b
rown","owner":"Shotaro"}},{"Key":"CAR99","Record":{"make":"ZCXY","model
":"XX","colour":"white","owner":"Cxy"}},{"Key":"PB19000046","Record":{"
make":"make","model":"model","colour":"blue","owner":"yyc"}},{"Key":"PB
19000046-2","Record":{"make":"make","model":"model","colour":"blue","ow
ner":"yyc"}},{"Key":"PB19000071","Record":{"make":"PengYiTeng","model":
"BlockChain","colour":"Lab4","owner":"Modify"}},{"Key":"PB19000078","Re
```



```
cord":{"make":"abc","model":"acb","colour":"green","owner":"wangzhen000
0"}},{"Key":"ZYM","Record":{"make":"zym","model":"zym","colour":"zym","
owner":"zym"}},{"Key":"car_test1","Record":{"make":"abc","model":"abc",
"colour":"white","owner":"temp2"}}]
bash-5.0# 
```

> B档（合并在A档中）

- 由于官方的fabcar中已经实现了增、改、查操作
- 故B档在官方fabcar的基础上加入了删除操作
- 具体实现见下

```go
// DeleteCar deletes the car found in world state
func (s *SmartContract) DeleteCar(ctx contractapi.TransactionContextInterface
, carNumber string) error {
        carAsBytes, err := ctx.GetStub().GetState(carNumber)

        if err != nil {
                return fmt.Errorf("Failed to read from world state. %s", err.Error())
        }

        if carAsBytes == nil {
                return fmt.Errorf("%s does not exist", carNumber)
        }

        return ctx.GetStub().DelState(carNumber)

    }
```

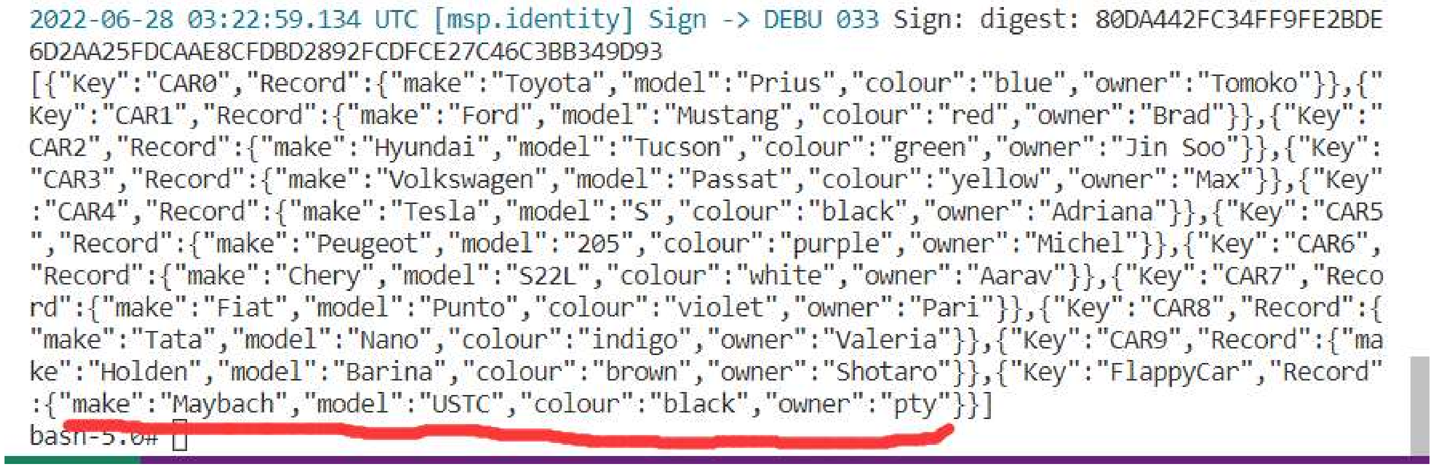
- 重新进行链码部署操作

```
 - peer lifecycle chaincode package fabcar_v1.tar.gz --path /etc/hyperledger/org1/chaincode/fabcar_v1/go/ --lang golang --label fabcar_2

 - export CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/org1/admin/msp

 - peer lifecycle chaincode install fabcar_v1.tar.gz

 - peer lifecycle chaincode queryinstalled


  export CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/org1/admin/msp


  export VERSION=1

  export PACKAGE_ID=fabcar_2:b89522b9c8240a06145ca2b4946bb27ff4391ea77ea2f8dc7b786a71f155b9ce

  export ORDERER_CA=/etc/hyperledger/org1/peer2/tls-msp/tlscacerts/tls-172-16-4-35-7052.pem

  export CHANNEL_NAME=mychannel


  peer lifecycle chaincode approveformyorg -o orderer1-org0:7050 --ordererTLSHostnameOverride orderer1-org0 --tls --cafile /etc/hyperledger/org1/peer2/tls-msp/tlscacerts/tls-172-16-4-35-7052.pem --

 peer lifecycle chaincode checkcommitreadiness  --channelID mychannel --name fabcar_v1 --version 1  --sequence 2 --output json

 peer lifecycle chaincode commit -o orderer1-org0:7050 --ordererTLSHostnameOverride orderer1-org0 --tls --cafile /etc/hyperledger/org1/peer2/tls-msp/tlscacerts/tls-172-16-4-35-7052.pem --channelID
```

- 启动链码并进行增删改查操作

增：

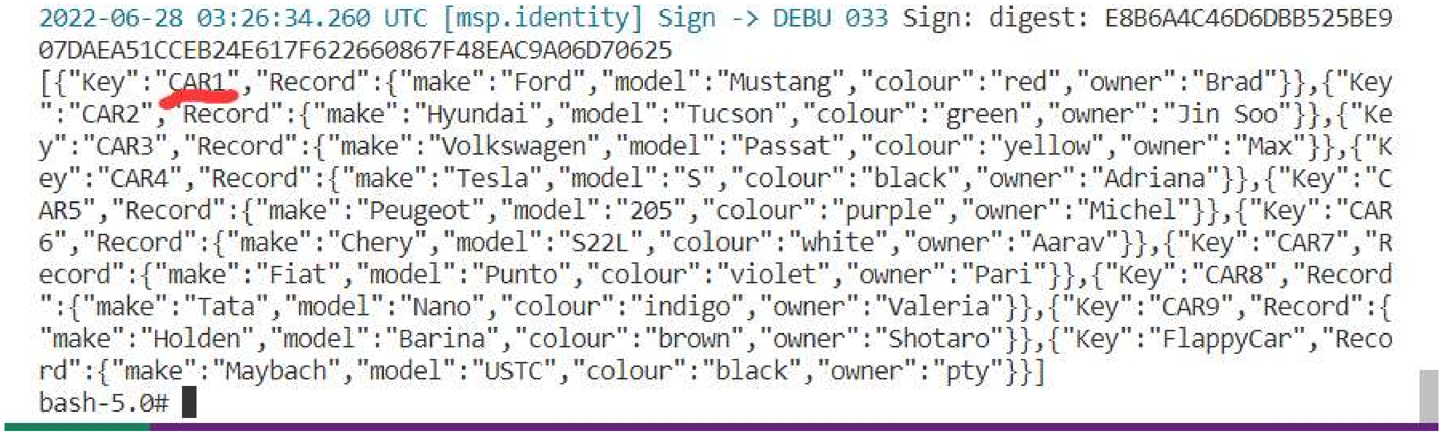
```
peer chaincode invoke -o orderer1-org0:7050 --ordererTLSHostnameOverride orderer1-org0 --tls --cafile /etc/hyperledger/org1/peer2/tls-msp/tlscacerts/tls-172-16-4-35-7052.pem -C mychannel -n fabcar_
```



```
2022-06-28 03:22:59.134 UTC [msp.identity] Sign -> DEBU 033 Sign: digest: 80DA442FC34FF9FE2BDE
6D2AA25FDCAAE8CFDBD2892FCDFCE27C46C3BB349D93
[{"Key":"CAR0","Record":{"make":"Toyota","model":"Prius","colour":"blue","owner":"Tomoko"}},{"
Key":"CAR1","Record":{"make":"Ford","model":"Mustang","colour":"red","owner":"Brad"}},{"Key":"
CAR2","Record":{"make":"Hyundai","model":"Tucson","colour":"green","owner":"Jin Soo"}},{"Key":
"CAR3","Record":{"make":"Volkswagen","model":"Passat","colour":"yellow","owner":"Max"}},{"Key"
:"CAR4","Record":{"make":"Tesla","model":"S","colour":"black","owner":"Adriana"}},{"Key":"CAR5
","Record":{"make":"Peugeot","model":"205","colour":"purple","owner":"Michel"}},{"Key":"CAR6",
"Record":{"make":"Chery","model":"S22L","colour":"white","owner":"Aarav"}},{"Key":"CAR7","Reco
rd":{"make":"Fiat","model":"Punto","colour":"violet","owner":"Pari"}},{"Key":"CAR8","Record":{
"make":"Tata","model":"Nano","colour":"indigo","owner":"Valeria"}},{"Key":"CAR9","Record":{"ma
ke":"Holden","model":"Barina","colour":"brown","owner":"Shotaro"}},{"Key":"FlappyCar","Record"
:{"make":"Maybach","model":"USTC","colour":"black","owner":"pty"}}]
bash-5.0# 
```

删：

```
peer chaincode invoke -o orderer1-org0:7050 --ordererTLSHostnameOverride orderer1-org0 --tls --cafile /etc/hyperledger/org1/peer2/tls-msp/tlscacerts/tls-172-16-4-35-7052.pem -C mychannel -n fabcar_
```



```
2022-06-28 03:26:34.260 UTC [msp.identity] Sign -> DEBU 033 Sign: digest: E8B6A4C46D6DBB525BE9
07DAEA51CCEB24E617F622660867F48EAC9A06D70625
[{"Key":"CAR1","Record":{"make":"Ford","model":"Mustang","colour":"red","owner":"Brad"}},{"Key
":"CAR2","Record":{"make":"Hyundai","model":"Tucson","colour":"green","owner":"Jin Soo"}},{"Ke
y":"CAR3","Record":{"make":"Volkswagen","model":"Passat","colour":"yellow","owner":"Max"}},{"K
ey":"CAR4","Record":{"make":"Tesla","model":"S","colour":"black","owner":"Adriana"}},{"Key":"C
AR5","Record":{"make":"Peugeot","model":"205","colour":"purple","owner":"Michel"}},{"Key":"CAR
6","Record":{"make":"Chery","model":"S22L","colour":"white","owner":"Aarav"}},{"Key":"CAR7","R
ecord":{"make":"Fiat","model":"Punto","colour":"violet","owner":"Pari"}},{"Key":"CAR8","Record
":{"make":"Tata","model":"Nano","colour":"indigo","owner":"Valeria"}},{"Key":"CAR9","Record":{
"make":"Holden","model":"Barina","colour":"brown","owner":"Shotaro"}},{"Key":"FlappyCar","Reco
rd":{"make":"Maybach","model":"USTC","colour":"black","owner":"pty"}}]
bash-5.0# 
```

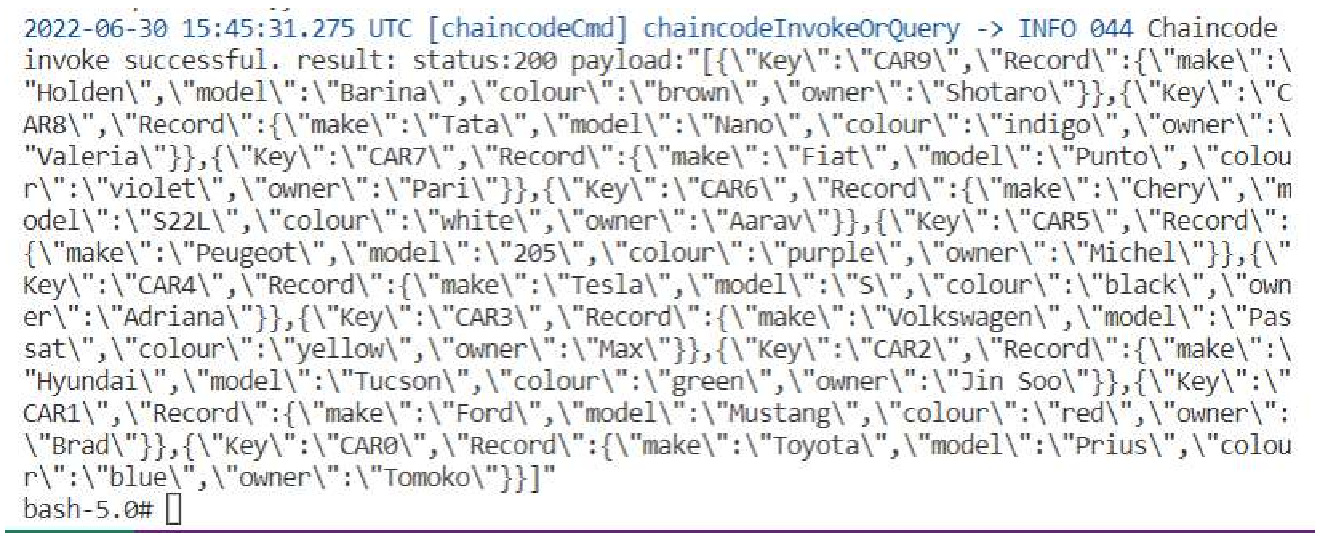
- 改和查这里不做演示

> A档（文件：fabcar_v2）

- 在B档的基础上实现按照Key的字典序的排序操作
- 代码如下

```
type SortQue []*QueryResult

func (s SortQue) Len() int {
                return len(s)
}
func (s SortQue) Swap(i, j int) {
                s[i], s[j] = s[j], s[i]
}
func (s SortQue) Less(i, j int) bool {
                return strings.Compare(s[i].Key, s[j].Key) == 1
}

func (s *SmartContract) SortCar(ctx contractapi.TransactionContextInterface) ([]*QueryResult, error) {
                startKey := ""
                endKey := ""

                resultsIterator, err := ctx.GetStub().GetStateByRange(startKey, endKey)

                if err != nil {
                                return nil, nil
                }
                defer resultsIterator.Close()

                results := []*QueryResult{}

                for resultsIterator.HasNext() {
                                queryResponse, err := resultsIterator.Next()

                                if err != nil {
                                                return nil, nil
                                }

                                car := new(Car)
                                _ = json.Unmarshal(queryResponse.Value, car)

                                queryResult := QueryResult{Key: queryResponse.Key, Record: car}
                                results = append(results, &queryResult)
                }
                sort.Sort(SortQue(results))

                return results, nil
```

排序后结果如下



```
2022-06-30 15:45:31.275 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 044 Chaincode
invoke successful. result: status:200 payload:"[{\"Key\":\"CAR9\",\"Record\":{\"make\":\
"Holden\",\"model\":\"Barina\",\"colour\":\"brown\",\"owner\":\"Shotaro\"}},{\"Key\":\"C
AR8\",\"Record\":{\"make\":\"Tata\",\"model\":\"Nano\",\"colour\":\"indigo\",\"owner\":\
"Valeria\"}},{\"Key\":\"CAR7\",\"Record\":{\"make\":\"Fiat\",\"model\":\"Punto\",\"colou
r\":\"violet\",\"owner\":\"Pari\"}},{\"Key\":\"CAR6\",\"Record\":{\"make\":\"Chery\",\"m
odel\":\"S22L\",\"colour\":\"white\",\"owner\":\"Aarav\"}},{\"Key\":\"CAR5\",\"Record\":
{\"make\":\"Peugeot\",\"model\":\"205\",\"colour\":\"purple\",\"owner\":\"Michel\"}},{\"
Key\":\"CAR4\",\"Record\":{\"make\":\"Tesla\",\"model\":\"S\",\"colour\":\"black\",\"own
er\":\"Adriana\"}},{\"Key\":\"CAR3\",\"Record\":{\"make\":\"Volkswagen\",\"model\":\"Pas
sat\",\"colour\":\"yellow\",\"owner\":\"Max\"}},{\"Key\":\"CAR2\",\"Record\":{\"make\":\
"Hyundai\",\"model\":\"Tucson\",\"colour\":\"green\",\"owner\":\"Jin Soo\"}},{\"Key\":\"
CAR1\",\"Record\":{\"make\":\"Ford\",\"model\":\"Mustang\",\"colour\":\"red\",\"owner\":
\"Brad\"}},{\"Key\":\"CAR0\",\"Record\":{\"make\":\"Toyota\",\"model\":\"Prius\",\"colou
r\":\"blue\",\"owner\":\"Tomoko\"}}]"
bash-5.0# 
```

一些感想

关于这次实验，我有十分多的话想说。我一共有三天时间完成这个实验，而且在一早就构思好想要实现的是基于属性访问控制的投票系统。利用节点证书属性的唯一性和不可篡改性，保证投票结果的公开透明，并保证了票数的数量可计算性。但事与愿违，我耗费了整整一天的时间也没有实现获取节点的属性信息。

而后，我想退而求次，实现匿名投票的投票系统。但却遇到了更加玄学的问题，这耗费了我一天半的时间去解决，但却无功而返。如果助教有时间，我希望助教能够运行一下我的代码。

无奈之下，我只能最后选择了在B的基础上实现排序。

最后，虽然最后两个实验让我痛苦了整整一周，但却学到了许多。（论悲催大学生被实验折磨的痛苦以及对应的收获，可以写小论文了，这里不再赘述）

最后的最后，感谢助教这几天给予我莫大的帮助，助教和老师都辛苦了！