

< 하이퍼레저 페브릭-4 >

2018. 11. 21(수)

1. marble_chaincode_private.go 추가 작성 후 업그레이드 하기

Q) marble03_private 디렉토리를 만들어 추가된 CC 작성 후 upgrade하기

- ① .json 파일 수정하기(collectionMarbleStatus 추가 / "Org2"만 가능하게끔)

```
{
  "name": "collectionMarbleStatus",
  "policy": "OR('Org2MSP.member')",
  "requiredPeerCount": 0,
  "maxPeerCount": 3,
  "blockToLive": 1000000
}
```

- ② chaincode 수정

- struct 추가(marbleStatusPDC) : Status값은 'stock', 'sold out', 'ready' 등등 값이 가능

```
type marbleStatusPDC struct {
    ObjectType string `json:"docType"`
    Name        string `json:"name"`
    Status      string `json:"status"`
}
```

- Invoke 함수에 case문 추가 후 "checkStatus" 함수 구현(func : readMarbles 그대로 복붙)

```
case "checkStatus":
    //check a marble status
    return t.checkStatus(stub, args)
```

: 말 그대로 readStatus하는 것(입력받은 값을 출력해주는 역할 할 수 있도록 함수 구현)

- initMarble 함수에 초기 값 하나 더 추가(5-price)

```
func (t *SimpleChaincode) initMarble(stub shim.ChaincodeStubInterface, args []string) pb.Response {
    var err error

    // 0-name 1-color 2-size 3-owner 4-price 5-price
    // "asdf", "blue", "35", "bob", "99", "stack"
    if len(args) != 6 {
        return shim.Error("Incorrect number of arguments. Expecting 6")
    }
}
```

: 즉, 하나 값을 더 받았기 때문에 한 번 더 쪼개줄 것(json 이용 PutStatus를 만들어주기)

A) 위에서 한 내용에 따라 코드 수정 후 업데이트!

① Invoke에서 select~case 문 작성한 것을 바탕으로 checkStatus라는 함수 생성

```
func (t *SimpleChaincode) checkStatus(stub shim.ChaincodeStubInterface, args []string) pb.Response {
    var name, jsonResp string
    var err error

    if len(args) != 1 {
        return shim.Error("Incorrect number of arguments. Expecting name of the marble to query")
    }

    name = args[0]
    valAsbytes, err := stub.GetPrivateData("collectionMarbleStatus", name) //get the marble from chaincode state
    if err != nil {
        jsonResp = "{\"Error\":\"Failed to get state for " + name + "\"}"
        return shim.Error(jsonResp)
    } else if valAsbytes == nil {
        jsonResp = "{\"Error\":\"Marble does not exist: " + name + "\"}"
        return shim.Error(jsonResp)
    }

    return shim.Success(valAsbytes)
}
```

: json에서 작성한 collectionMarbleStatus 함수를 가져와서 그 이름(get)에 해당하는 값을 반환(=status)
: 에러가 발생할 경우(null이 아닐 경우) status 받아오는데 문제 생긴 것
: 가져오려는 값(status)인 valAsbytes가 null일 경우 marble이 존재하지 않는 것
: 둘 다 이상이 없으면 response로 valAsbytes(=status) 값을 그대로 출력!!

② initMarble()에서 데이터 하나 추가해 총 6개 값을 받도록 했기 때문에 그 안에 내용 수정

```
if len(args[5]) == 0 {
    return shim.Error("6th argument must be a non-empty string")
}
marbleName := args[0]
color := strings.ToLower(args[1])
owner := strings.ToLower(args[3])
size, err := strconv.Atoi(args[2]) // 문자를 숫자로 변경할 때 쓰이는 Atoi는 반드시 2개의 값을 반환하므로 err도 써줄 것(저 값 안 쓸꺼면 _로)
if err != nil {
    return shim.Error("3rd argument must be a numeric string")
}
price, err := strconv.Atoi(args[4])
if err != nil {
    return shim.Error("5th argument must be a numeric string")
}
status := strings.ToLower(args[5]) // 생성과 동시에 할당(=)
```

: if args[5](=status)의 길이가 0이면 제대로 입력하라는 에러 문구 출력
: 입력받은 args[5]의 값을 status라는 변수로 지정(strings 타입으로 생성과 동시에 할당!)

```
objectType = "marbleStatus" // 단순 이름 지정이므로 struct타입과 꼭 같을 필요 없음
marbleStatus := &marbleStatusPDC{objectType, marbleName, status} // 구조체의 주소지정(&)
marbleStatusBytes, err := json.Marshal(marbleStatus)
if err != nil {
    return shim.Error(err.Error())
}
err = stub.PutPrivateData("collectionMarbleStatus", marbleName, marbleStatusBytes)
if err != nil {
    return shim.Error(err.Error())
}
```

: init()으로 입력받은 값을 collectionMarbleStatus에 저장하기 위해 PutPrivateData 구현
: marbleName에 해당하는 marbleStatusBytes값을 입력(Marshal함수를 통해 바이트형식으로 변환 후)

③ delete()에 status 삭제하는 코드 추가

```
// Delete private status of marble
err = stub.DelPrivateData("collectionMarbleStatus", marbleName)
if err != nil {
    return shim.Error(err.Error())
}
```

④ \$./byfn.sh down으로 모든 채널,서버 죽이고

\$./byfn.sh up -c mychannel -s couchdb를 이용해 couchdb를 갖는 새로운 시스템 기동 state(저장소)

: 하게 되면 channel 생성, join, anchor-peer 생성, chaincode install, instantiate까지 자동으로 실행!!
: 여기서 생성되는 체인코드는 sample_example02로 내장된 코드(byfn.sh에서 삭제해버리면 작동 안함)
: byfn.sh 파일은 네트워크 기동을 쉽게 해주기 위한 스크립트 파일일 뿐!!

\$ docker ps로 네트워크 잘 띄워졌는지 확인

\$ docker exec -it cli bash를 이용해 docker로 이동!!(이 안에서 install 등 작업!)
interpret terminal

\$ peer chaincode list —installed 와 peer chaincode list —instantiated -C mychannel

: 해보면 byfn.sh 파일에 내장되어 있는 기본값인 mycc만 설치되어 있는 것을 확인 가능
: 이건 별개로, marble관련 체인코드를 설치하고 인스턴스화 해줘야함!!

marblesp:1.0을 install하고, instantiate(강사님 github 참고!)

: down했으니까 어제 상태로 되돌리기 한 것
: Org1에서 price(보임!) Org2에서 price(에러!)

현재상황

install	marblesp : 1.0 mycc : 1.0
instance	marblesp : 1.0 mycc : 1.0

⑤ marblesp 2.0 install

\$ peer chaincode install -n marblesp -v 2.0 -p github.com/chaincode/marbles03_private/go
: 각 peer별로 환경변수 바뀌가며 install 작업 실시!(총 4번)

⑥ marblesp 2.0 upgrade

\$ export ORDERER_CA=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
\$ peer chaincode upgrade -o orderer.example.com:7050 --tls —cafile \$ORDERER_CA
-C mychannel -n marblesp -v 2.0 -c '{"Args":["init"]}' -P "OR('Org1MSP.member','Org2MSP.member')"
--collections-config \$GOPATH/src/github.com/chaincode/marbles03_private/collections_config.json

: instantiate를 upgrade로만 바꿔서 실행(기존 marblesp 1.0 인스턴스를 업그레이드 하는 작업)
: 이후 installed된 list 확인해보면 3개 나옴(mycc, marblesp 1.0, marblesp 2.0)
: 해당 채널에 instantiated된 list 확인해보면 2개 나옴(mycc, marblesp 2.0) => 1.0은 2.0으로 덮어쓰워짐

⑦ invoke(initMarble)

\$ peer chaincode invoke -o orderer.example.com:7050 —tls —cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n marblesp
-c '{"Args":["initMarble","marble2","blue","35","tom","99","stock"]}'

: 이제 upgrade 한 chaincode를 바탕으로 invoke(initMarble 함수 이용) 작업 실시
: 만약 "marble1"으로 생성하면 error발생(why? marblesp 1.0에서 이미 생성한 데이터이기 때문!)
: 그러므로, "marble2"로 바꾸고, 맨 마지막에 "stock"이라는 status 정보 추가해서 작성할 것(총 6개 argument)

⑧ query

\$ peer chaincode query -C mychannel -n marblesp -c '{"Args":["checkStatus","marble"]}'
: Org2에서 하면 정상작동! Org1로 바꿔 쿼리 날리면 미 작동!(json 파일에서 그렇게 설정했기 때문에)

```
root@af9ceab37b62:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C mychannel -n marblesp -c '{"Args":["checkStatus","marble"]}'
{"docType":"marbleStatus","name":"marble","status":"stock"}
root@af9ceab37b62:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C mychannel -n marblesp -c '{"Args":["checkStat
us","marble"]}'
Error: endorsement failure during query. response: status:500 message:{"Error":"Failed to get state for marble"}
```

현재상황

install	marblesp : 2.0 marblesp : 1.0 mycc : 1.0
instance	marblesp : 2.0 mycc : 1.0

2. 체인코드 basic-network로 시작하기

- fabric-samples/basic-network 디렉토리에 들어가보면 다양한 .sh파일이 있음
단순 스크립트 파일로 이를 사용해 간편하게 서버를 구동할 수 있음
모든 프로젝트를 시작할 때 이 basic-network를 이용해서 구현!!

```
guru@guru:~/fabric-samples/basic-network$ ls
config  configtx.yaml  crypto-config  crypto-config.yaml  docker-compose.yml  generate.sh  init.sh  README.md  start.sh  stop.sh  teardown.sh
```

- 모든 business logic은 전부 chaincode 안에 들어있어야 함!
- chaincode는 GetState, PutState, DelState라는 함수를 지원
- **설계과정 : 기능일람표 작성(어느 기능이 필요한지) → API로 바꿈 → 구현**
- 네트워크가 안될 땐??? \$ **sudo service networking restart**로 네트워크 재시작!

```
guru@guru:~$ sudo service networking restart
```

2. tuna 실습

- home에 dir만들고 basic-network에 있는 것 전체를 복사(\$ cp -rf ./basic-network/ ~/tuna/)

```
guru@guru:~$ tree tuna -L 1
tuna
├── basic-network
├── chaincode
└── tuna-app
```

: 앞으로 fabric 프로젝트 할 때는 basic-network 안에서 모두 실행!!

```
guru@guru:~/tuna/basic-network$ ll
total 64
drwxrwxr-x 4 guru guru 4096 Nov 21 17:15 ./
drwxrwxr-x 5 guru guru 4096 Nov 21 17:12 ../
drwxrwxr-x 2 guru guru 4096 Nov 21 17:15 config/
-rw-rw-r-- 1 guru guru 4341 Nov 21 16:36 configtx.yaml
drwxrwxr-x 4 guru guru 4096 Nov 21 17:15 crypto-config/
-rw-rw-r-- 1 guru guru 3555 Nov 21 16:36 crypto-config.yaml
-rw-rw-r-- 1 guru guru 4774 Nov 21 16:36 docker-compose.yml
-rw-rw-r-- 1 guru guru 25 Nov 21 16:36 .env
-rwxrwxr-x 1 guru guru 1238 Nov 21 17:15 generate.sh*
-rwxrwxr-x 1 guru guru 277 Nov 21 16:36 init.sh*
-rw-rw-r-- 1 guru guru 770 Nov 21 16:36 README.md
-rwxrwxr-x 1 guru guru 1116 Nov 21 16:36 start.sh*
-rwxrwxr-x 1 guru guru 258 Nov 21 16:36 stop.sh*
-rwxrwxr-x 1 guru guru 468 Nov 21 16:36 teardown.sh*
```

: ./teardown.sh(중지하고 죽이는 스크립트)

: ./start.sh(네트워크 기동하는 스크립트 : endorsor만들기 등등

: 실제 시작은 ./generate.sh로 시작!!

Branch: master ▾


education / LFS171x / fabric-material / tuna-app /

Create new file

Upload files


Find file

History

 alexandrag2254 and tkuhr Add attributes (#8) ...


Latest commit 6b43283 on 27 Oct 2017

..

 client


Add Hyperledger Fabric code to Hyperledger education repo (#7)

a year ago

 src


Add attributes (#8)

a year ago

 .gitignore


Add Hyperledger Fabric code to Hyperledger education repo (#7)

a year ago

 README.md


Add attributes (#8)

a year ago

 controller.js


Add attributes (#8)

a year ago

 package-lock.json


Add Hyperledger Fabric code to Hyperledger education repo (#7)

a year ago

 package.json


Add Hyperledger Fabric code to Hyperledger education repo (#7)

a year ago

 registerAdmin.js


Add attributes (#8)

a year ago

 registerUser.js


Add attributes (#8)

a year ago

 routes.js


Add Hyperledger Fabric code to Hyperledger education repo (#7)

a year ago

 server.js

Add Hyperledger Fabric code to Hyperledger education repo (#7)

a year ago

 startFabric.sh

Add attributes (#8)

a year ago

오류 메시지의 출력 단계(←단계까지 다 보여줌 : ex)debug라고 바꾸면 verbose, info에러 보여줌)
 verbose / info / debug / warn / error / fatal

```
cli:
  container_name: cli
  image: hyperledger/fabric-tools
  tty: true
  environment:
    - GOPATH=/opt/gopath
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    - CORE_LOGGING_LEVEL=info
    - CORE_PEER_ID=cli
    - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
    - CORE_PEER_LOCALMSPID=Org1MSP
```

이런걸 info 지우고 debug로 수정!

```
2018-11-21 08:49:21.824 UTC [grpc] HandleSubConnStateChange -> DEBU 03c pickfirstBalancer: HandleSubConnStateChange: 0xc4205e89e0, READY
2018-11-21 08:49:21.828 UTC [grpc] DialContext -> DEBU 03d parsed scheme: ""
2018-11-21 08:49:21.829 UTC [grpc] DialContext -> DEBU 03e scheme "" not registered, fallback to default scheme
2018-11-21 08:49:21.830 UTC [grpc] watcher -> DEBU 03f ccResolverWrapper: sending new addresses to cc: [{peer0.org1.example.com:7051 0 <nil>}]
2018-11-21 08:49:21.830 UTC [grpc] switchBalancer -> DEBU 040 ClientConn switching balancer to "pick_first"
2018-11-21 08:49:21.831 UTC [grpc] HandleSubConnStateChange -> DEBU 041 pickfirstBalancer: HandleSubConnStateChange: 0xc420068c50, CONNECTING
2018-11-21 08:49:21.833 UTC [grpc] HandleSubConnStateChange -> DEBU 042 pickfirstBalancer: HandleSubConnStateChange: 0xc420068c50, READY
2018-11-21 08:49:21.834 UTC [msp] GetDefaultSigningIdentity -> DEBU 043 Obtaining default signing identity
2018-11-21 08:49:21.835 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 044 Using default escc
2018-11-21 08:49:21.835 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 045 Using default vscc
Error: error getting chaincode code tuna-app: path to chaincode does not exist: /opt/gopath/src/github.com/tuna-app
```

에러가 더 많이 발생함!(하늘색 추가생성)


```

"name": "Fabric-tuna-app",
"version": "1.0.0",
"description": "Hyperledger Fabric MOOC Sample Application",
"main": "server.js",
"dependencies": {
  "fabric-ca-client": "^1.0.2",
  "fabric-client": "^1.0.2",
  "grpc": "^1.6.0",
  "express": "latest",
  "body-parser": "latest",
  "ejs": "latest",
  "angular": "^1.4.3"
},
"license": "Apache-2.0",
"keywords": [
  "Hyperledger",
  "Fabric",

```

package.json 파일

fabric-ca-client, fabric-client, grpc가 SDK역할! 애네에 필요한걸 다운받아주는 명령어가
\$ npm install 알아서 자동설치가 되면?

```

guru@guru:~/tuna/tuna-app$ ll
total 292
drwxrwxr-x   3 guru guru   4096 Nov 21 17:54 ./
drwxrwxr-x   5 guru guru   4096 Nov 21 17:12 ../
-rw-rw-r--   1 guru guru  20935 Nov 21 17:37 controller.js
drwxrwxr-x 346 guru guru  12288 Nov 21 17:54 node_modules/
-rw-rw-r--   1 guru guru    579 Nov 21 17:37 package.json
-rw-rw-r--   1 guru guru 228240 Nov 21 17:54 package-lock.json
-rw-rw-r--   1 guru guru   2939 Nov 21 17:37 registerAdmin.js
-rw-rw-r--   1 guru guru   3261 Nov 21 17:37 registerUser.js
-rw-rw-r--   1 guru guru    462 Nov 21 17:37 routes.js
-rw-rw-r--   1 guru guru   1204 Nov 21 17:37 server.js
-rwxrwxr-x   1 guru guru   1820 Nov 21 17:37 startFabric.sh*

```

node_modules 디렉토리가 자동 생성(들어가보면 굉장히 많은 파일들이 생성됨)

그리고 포트포워딩 8000번으로 실시!!