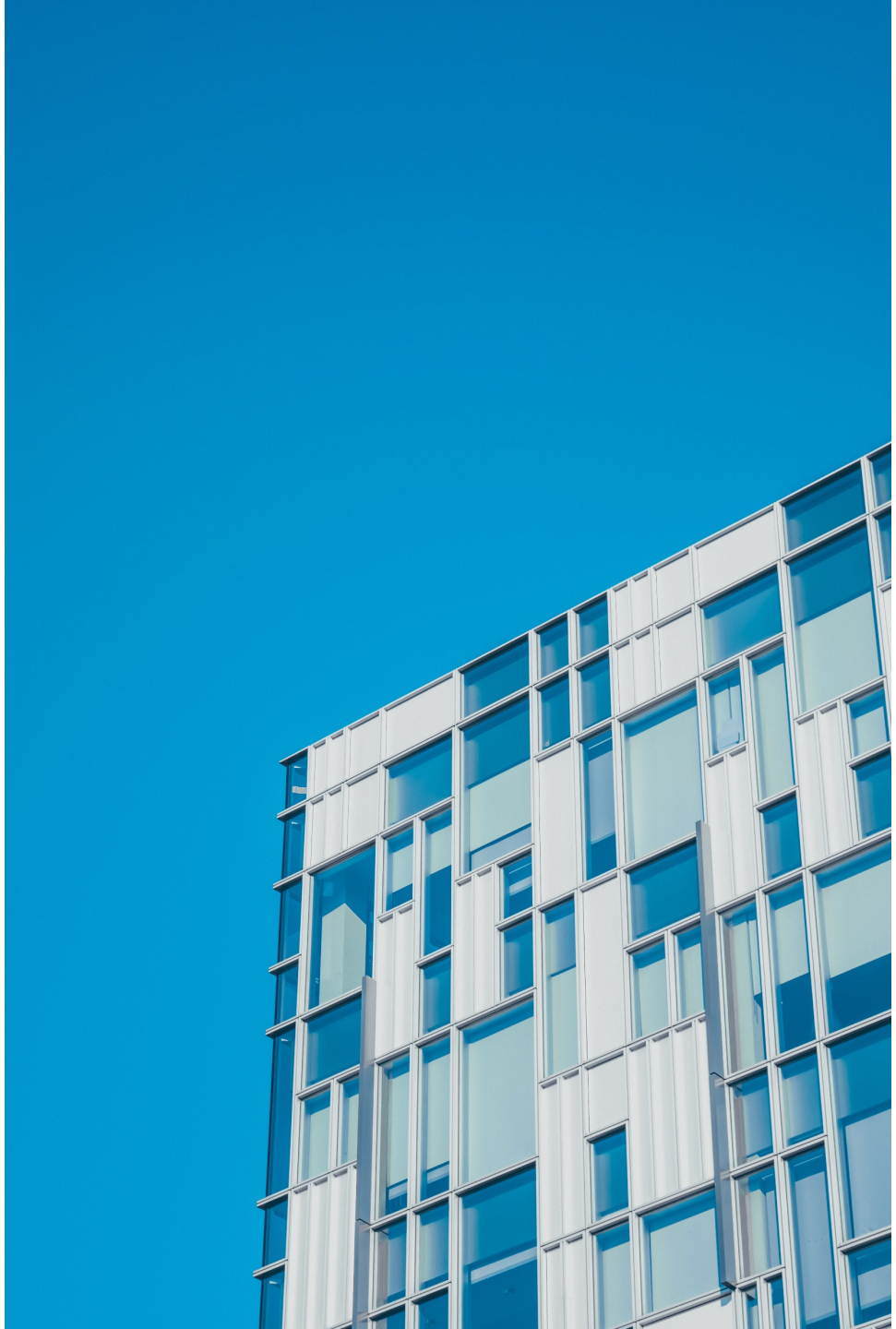
Table of Content



PART 1

Project

Project

중고차를 거래하는 시스템

- 사용자들은 blockchain에 사용자 등록을 합니다.
- 사용자들은 자신의 차량을 blockchain에 등록 할 수 있습니다.
- 사용자들은 등록한 자신의 차량을 판매가와 함께 판매 등록 할 수 있습니다.
- 사용자들은 판매로 올라온 차량을 구매할 수 있습니다.
- 판매 차량이 done인 경우 구매 차량 목록에 표시하지 않습니다.
- Sale중인 차량의 경우 판매 차량 목록에 표시하지 않습니다.
- 등록 차량, 판매 차량을 조회 할 수 있습니다.

Project

- 제출 목록 : 소스 코드 및 설계 문서 파일

  소스코드는 node_module를 제외한 폴더 및 파일을 포함하여 jar/zip 형태로 압축하여 "학번_이름_assignment2.jar/zip"로 제출

- 기한 : 12월 10일 화요일 23:59 까지 온라인에 소스코드 제출

  12월 11일 수요일 수업 시작 전 설계 문서 파일 제출

- 패널티 : 11일 수요일 23:59 이내(24 시간 이내)에 제출한 경우 50% 인정

  12일 금요일 23:59 이내(48 시간 이내)에 제출한 경우 25% 인정

  이후의 제출은 인정하지 않습니다.

PART 2

Fabric

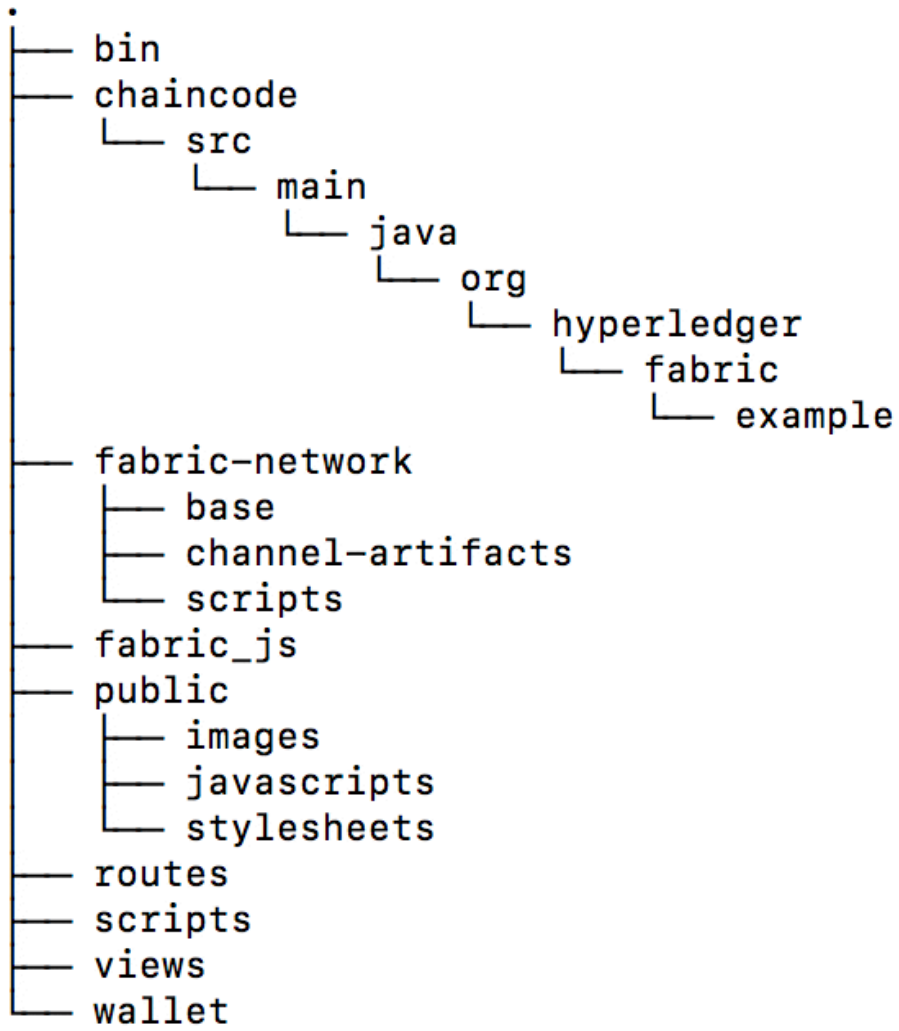
Fabric

- Fabric network를 만들기 위해 오른쪽과 같은 파일이 제공됩니다.

- Node를 사용하여 웹서버를 실행하며 fabric_js폴더의 node sdk를 사용하여 fabric network와 통신합니다.

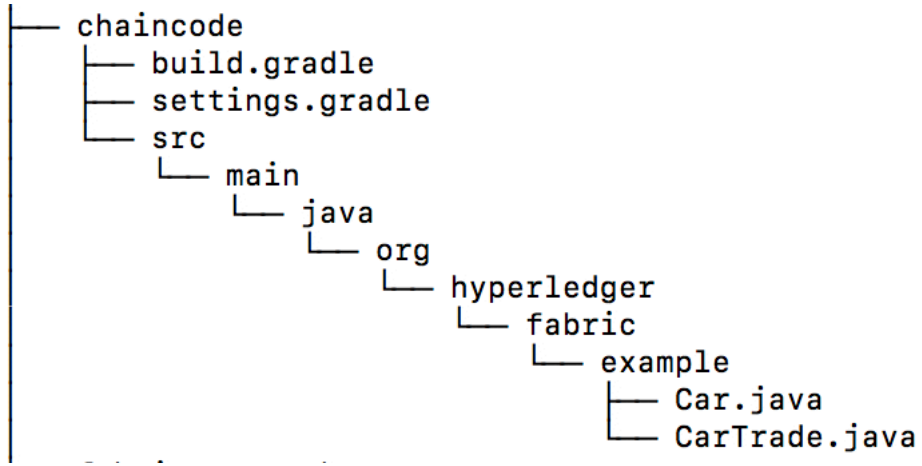
- enrollAdmin : admin 계정 생성
- registerUser : 사용자 계정 생성
- invoke : transaction 실행
- query : query 실행

- bin폴더는 바이너리 파일이 들어있습니다.
- App.js 는 서버를 실행시키는 파일입니다.
- routes/index.js는 요청 라우터 파일입니다.
- views/index.ejs 파일은 사용자 UI를 나타내는 파일입니다.
- scripts/fabric.sh 파일은 test에 사용되는 명령어 모음 파일입니다.
- 제공된 파일을 다운받은 후 npm install로 패키지 다운로드
- npm start로 서버 실행



Fabric

- Car.java코드
- 위치



- Car객체

-> make : 제조사

-> model : 모델명

-> color : 색상

-> owner : 차량 소유자 이름

```
public Car(final String make, final String model, final String color, final String owner) {
    this.make = make;
    this.model = model;
    this.color = color;
    this.owner = owner;
}
```

Fabric

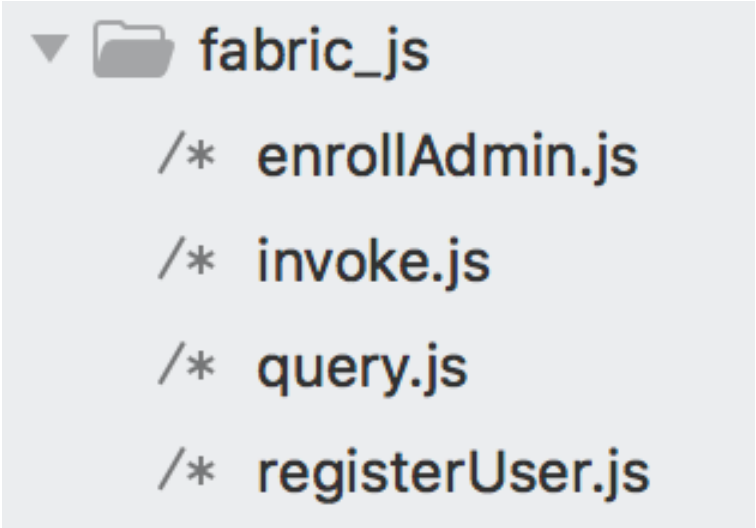
- CarTrade.java 코드

- registerCar : 차량 등록 함수
- sellMyCar : 자신의 차량 판매 등록 함수
- buyUserCar : 판매 등록된 차량 구매 함수
- changeCarOwner : 차량 소유자 변경 함수
- getMyCar : 자신의 차량 조회
- getAllRegisteredCar : 모든 차량 조회
- getAllOrderedCar : 판매(진행/완료) 차량 조회

```java
@Override
public Response invoke(ChaincodeStub stub) {
    try {
        _logger.info("Invoke java chaincode");
        String func = stub.getFunction();
        List<String> params = stub.getParameters();
        if (func.equals("registerCar")) {
            return registerCar(stub, params);
        }
        if (func.equals("sellMyCar")) {
            return sellMyCar(stub, params);
        }
        if (func.equals("buyUserCar")) {
            return buyUserCar(stub, params);
        }
        if (func.equals("changeCarOwner")) {
            return changeCarOwner(stub, params);
        }
        if (func.equals("getMyCar")) {
            return getMyCar(stub, params);
        }
        if (func.equals("getAllRegisteredCar")) {
            return getAllRegisteredCar(stub, params);
        }
        if (func.equals("getAllOrderedCar")) {
            return getAllOrderedCar(stub, params);
        }
```

Fabric

- Fabric node sdk 코드 작성
- Invoke.js, query.js 코드 작성
- Fabcar 코드를 참고하시어 transaction 생성 및 쿼리를 result에 담아 return 합니다.

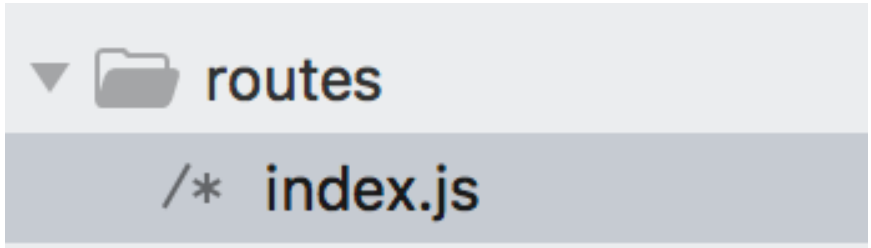



```
const result = await contract.evaluateTransaction('queryCar', 'CAR12');
```

```
await contract.submitTransaction('createCar', 'CAR12', 'Honda', 'Accord', 'Black', 'Tom');
```

Fabric

- Sdk와 통신하는 index.js 라우터




```
var enroll = require('../fabric_js/enrollAdmin.js')
var register = require('../fabric_js/registerUser.js')
var query = require('../fabric_js/query.js')
var invoke = require('../fabric_js/invoke.js')
```

4개의 node sdk 파일을
각 변수에 정의합니다.

```
router.post('/registerUser', function(req, res, next) {
    user = req.body.user;
    console.log(user);
    register.registerUser(user);
    res.cookie('user', user);
    res.redirect('/');
})
```

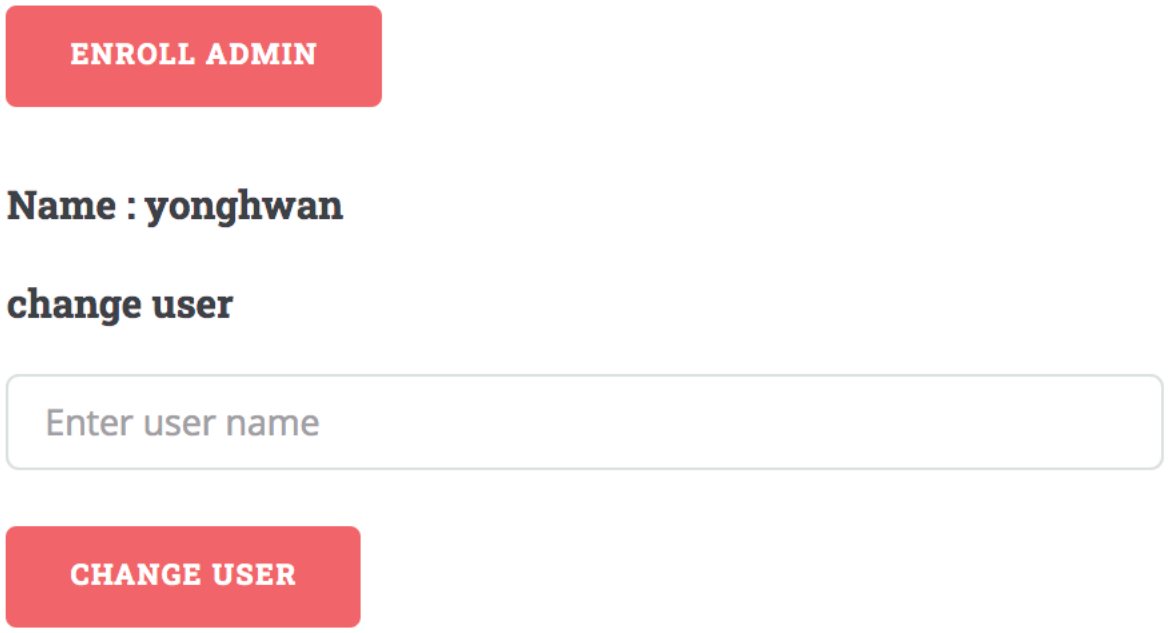
함수 사용법은 위에서 정의한 변수.
함수()로 사용합니다.
Ex) register.registerUser(user);

Fabric

- Identity
- Enroll Admin 버튼을 클릭하면 fabric_js 폴더의 enrollAdmin.js 파일이 실행되며, admin 계정이 wallet에 생성됩니다.
- Change user 버튼을 클릭하면 fabric_js 폴더의 registerUser.js 파일이 실행되며, user계정이 wallet에 생성됩니다.
- Name 필드에는 사용자 이름이 항상 표시되어야 합니다.

* 위 기능은 구현이 되어있으므로 따로 구현하실 필요는 없습니다. Invoke, query기능을 구현하실 때 소스코드 참고 바랍니다.

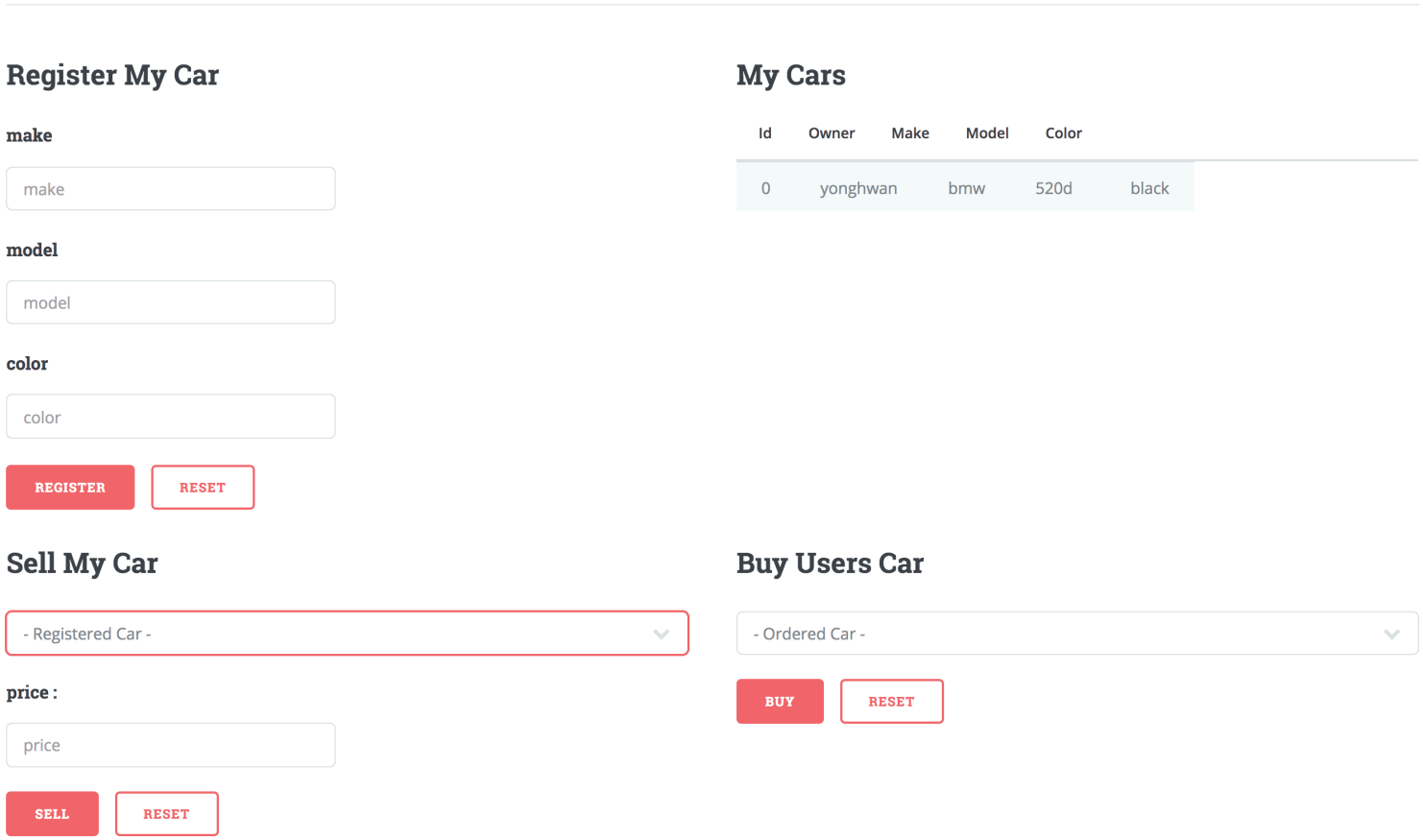
Idendity



Fabric

- Dapp -> Register My Car, My Cars, Sell My Car, Buy Users Car

- 판매 차량이 done인 경우 Buy Users Car에 표시하지 않으며, Sale중인 차량의 경우 Sell My Car에 표시하지 않습니다.
- 차량은 사용자마다 1대를 소유할 수 있다고 가정합니다.



Fabric

- Dapp -> registered cars, cars on sale
- 등록된 모든 차량과 판매(진행, 완료) 차량을 표시합니다.

---

registered cars

Id	Owner	Make	Model	Color

cars on sale

Id	Owner	Make	Model	Color	Price	Status

PART 3

Test

Test

- Test에서 진행하는 명령어는 scripts/fabric.sh에서 확인 바랍니다.
- url : http://dcc.hanyang.ac.kr/
- dcc -> Announcements ->Programming Assignment #2 -> fabric_project_interface.zip 파일 다운로드

Test

- 압축 해제
- 실습에서 진행한 fabric -> fabric_samples 폴더에서 bin 폴더를 fabric_project_interface/bin 폴더로 복사해줍니다.

```
iamyonghwan-MacBook-Pro:fabric-samples iamyonghwan$ pwd
/Users/iamyonghwan/fabric/fabric-samples
iamyonghwan-MacBook-Pro:fabric-samples iamyonghwan$ ls
CODE_OF_CONDUCT.md              MAINTAINERS.md                  basic-network
CONTRIBUTING.md                 README.md                       bin
Jenkinsfile                     SECURITY.md                     chaincode
LICENSE                         balance-transfer                chaincode-docker-devmode


iamyonghwan-MacBook-Pro:fabric-samples iamyonghwan$ cp -r bin/ ~/fabric_project_interface/bin
```

Test

- Docker images 명령어로 다음과 같이 이미지가 설치 되어있는 지를 확인합니다.
- Docker ps 명령어로 container가 없는지 확인합니다.

```
iamyonghwan-MacBook-Pro:fabric-samples iamyonghwan$ docker images
REPOSITORY                     TAG                 IMAGE ID            CREATED             SIZE
hyperledger/fabric-javaenv     1.4.4               4648059d209e        2 weeks ago         1.7GB
hyperledger/fabric-javaenv     amd64-1.4.4         4648059d209e        2 weeks ago         1.7GB
hyperledger/fabric-javaenv     latest              4648059d209e        2 weeks ago         1.7GB
hyperledger/fabric-ca          1.4.4               62a60c5459ae        2 weeks ago         150MB
hyperledger/fabric-ca          latest              62a60c5459ae        2 weeks ago         150MB
hyperledger/fabric-tools       1.4.4               7552e1968c0b        2 weeks ago         1.49GB
hyperledger/fabric-tools       latest              7552e1968c0b        2 weeks ago         1.49GB
hyperledger/fabric-ccenv       1.4.4               ca4780293e4c        2 weeks ago         1.37GB
hyperledger/fabric-ccenv       latest              ca4780293e4c        2 weeks ago         1.37GB
hyperledger/fabric-orderer     1.4.4               dbc9f65443aa        2 weeks ago         120MB
hyperledger/fabric-orderer     latest              dbc9f65443aa        2 weeks ago         120MB
hyperledger/fabric-peer        1.4.4               9756aed98c6b        2 weeks ago         128MB
hyperledger/fabric-peer        latest              9756aed98c6b        2 weeks ago         128MB
hyperledger/fabric-zookeeper   0.4.18              ede9389347db        3 weeks ago         276MB
hyperledger/fabric-zookeeper   latest              ede9389347db        3 weeks ago         276MB
hyperledger/fabric-kafka       0.4.18              caaae0474ef2        3 weeks ago         270MB
hyperledger/fabric-kafka       latest              caaae0474ef2        3 weeks ago         270MB
hyperledger/fabric-couchdb     0.4.18              d369d4eaa0fd        3 weeks ago         261MB
hyperledger/fabric-couchdb     latest              d369d4eaa0fd        3 weeks ago         261MB
hyperledger/fabric-baseimage   amd64-0.4.18        9e353eca480f        3 weeks ago         1.3GB
hyperledger/fabric-baseos      amd64-0.4.18        c256a6aad46f        3 weeks ago         80.8MB
iamyonghwan-MacBook-Pro:fabric-samples iamyonghwan$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
```

Test

- 컨테이너가 실행되었을 경우

-> fabric-network 폴더로 이동합니다.

-> ./byfn –m down 명령어로 컨테이너를 지워줍니다.

```
iamyonghwan-MacBook-Pro:fabric_project iamyonghwan$ cd fabric-network/
iamyonghwan-MacBook-Pro:fabric-network iamyonghwan$ ./byfn.sh -m down
Stopping for channel 'mychannel' with CLI timeout of '10' seconds and CLI delay of '3' seconds
Continue? [Y/n] Y
proceeding ...
```

Test

- 네트워크 실행
- fabric-network 폴더로 이동하여 ./byfn.sh –m up –a –n –s couchdb 명령어를 실행합니다.
- -a : ca 생성, -n : 체인코드 미설치, -s : state db 설정

Test

- docker ps 명령어로 다음과 같이 컨테이너가 실행되었는지 확인합니다.

```
iamyonghwan-MacBook-Pro:fabric-network iamyonghwan$ docker ps
CONTAINER ID        IMAGE                               COMMAND                  CREATED             STATUS              PORTS                                        NAMES
1f3ba721c64d        hyperledger/fabric-tools:latest     "/bin/bash"              36 seconds ago      Up 34 seconds                                                    cli
fe478b56e8b6        hyperledger/fabric-peer:latest      "peer node start"        37 seconds ago      Up 35 seconds       0.0.0.0:8051->8051/tcp                       peer1.org1.example.com
c546fd3c59a6        hyperledger/fabric-peer:latest      "peer node start"        39 seconds ago      Up 35 seconds       0.0.0.0:7051->7051/tcp                       peer0.org1.example.com
74a190531e7c        hyperledger/fabric-peer:latest      "peer node start"        39 seconds ago      Up 35 seconds       0.0.0.0:9051->9051/tcp                       peer0.org2.example.com
4892b8fb63b4        hyperledger/fabric-peer:latest      "peer node start"        39 seconds ago      Up 36 seconds       0.0.0.0:10051->10051/tcp                     peer1.org2.example.com
99835bd7c7c8        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   41 seconds ago      Up 38 seconds       4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp   couchdb0
b47ea307e893        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   41 seconds ago      Up 37 seconds       4369/tcp, 9100/tcp, 0.0.0.0:6984->5984/tcp   couchdb1
e19a39b9229f        hyperledger/fabric-orderer:latest   "orderer"                41 seconds ago      Up 37 seconds       0.0.0.0:7050->7050/tcp                       orderer.example.com
829e73baefb3        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   41 seconds ago      Up 39 seconds       4369/tcp, 9100/tcp, 0.0.0.0:8984->5984/tcp   couchdb3
e52f2043e978        hyperledger/fabric-ca:latest        "sh -c 'fabric-ca-se…"   41 seconds ago      Up 36 seconds       7054/tcp, 0.0.0.0:8054->8054/tcp             ca_peerOrg2
1f983fed9feb        hyperledger/fabric-ca:latest        "sh -c 'fabric-ca-se…"   41 seconds ago      Up 38 seconds       0.0.0.0:7054->7054/tcp                       ca_peerOrg1
91d212d3f201        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   41 seconds ago      Up 38 seconds       4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp   couchdb2
```

Test

- Cli 컨테이너 접속, chaincode install, instantiate, invoke query

```
# execute command line interface container
docker exec -it cli bash

# chaincode install
peer chaincode install -n mycc -v 1.0 -l java -p /opt/gopath/src/github.com/chaincode/

# check chaincode install
peer chaincode list --installed

# instantiate chaincode
peer chaincode instantiate -o orderer.example.com:7050 --tls --cafile $ORDERER_CA -C $

# invoke chaincode
peer chaincode invoke -o orderer.example.com:7050 --tls --cafile $ORDERER_CA -C $CHANNE

# query chaincode
peer chaincode query -C mychannel -n mycc -c '{"Args":["query","a"]}'
```

그 밖에 문의 사항에 대해서는, 010-9177-1204 또는 iamyonghwan@nate.com 으로 연락 주시기 바랍니다.