

## 주간 연구보고서

작성자	김한호	작성일자	2020.04.19.
연구기간	2020.04.13 ~ 2020.04.19		
1. 주간 연구 요약			
hyperledger 간단한 부동산 예제 프로그램 작성			
• realestatetransfer			
Participant: 거래 참여자			
Realestate: 거래되는 부동산			
• 테스트			
testing 패키지를 통해서 테스트			
2. 주간 연구 상세			
저번 주에 예제 프로그램을 수정해서 간단하게 개발하는 것이 나올 것 같다고 이야기하셔서 fabcar를 참조해서 프로그램을 작성했습니다. basic-network를 복사해서 configtx.yaml을 편집해서 혹시나 있을 org 갯수가 늘어날 것을 생각해서 3개의 org로 만들었습니다. crypto-config.yaml도 위에 변경사항에 맞춰서 수정을 했습니다. generate.sh도 추가된 org에 맞추어서 org1에 있는 내용을 복사해서 org2, org3를 추가했습니다. 다음으로 암호와 인증키를 생성했습니다. 다음으로 각 생성된 조직의 CA비밀키 파일명을 확인했습니다. 다음으로 docker-compose.yaml을 편집해서 CA 컨테이너 피어 컨테이너에 Org2, Org3를 추가하고 CA 관리자 패스워드를 테스트가 편하게 수정했습니다. TLS 암호화 통신을 활성화했습니다. 블록체인에 관한 내용을 구현하는 것이 아니라 연구를 하는 것이라고 생각해서 CouchDB가 아니라 LevelDB를 사용했습니다. 그리고 이후 CLI 컨테이너를 구성했습니다. 다음으로 start.sh와 teardown.sh를 위에 변경된 내용에 따라 편집했습니다. 블록체인 네트워크를 기동하고 실제 네트워크가 동작을 하는지 docker exec cli peer channel list로 동작을 확인했습니다. 다음으로 응용 프로그램을 개발했습니다.			

```

type Participant struct {
    Id string
    Name string
    Token string
}

type Realestate struct {
    Id string
    Name string
    OwnerId string
    IsTrading string
    Timestamp time.Time
}

type RealestateTransfer interface {
    AddParticipant(shim.ChaincodeStubInterface, *Participant) error
    CheckParticipant(shim.ChaincodeStubInterface, string) (bool, error)
    ListParticipants(shim.ChaincodeStubInterface) ([]*Participant, error)

    AddRealestate(shim.ChaincodeStubInterface, *Realestate) error
    CheckRealestate(shim.ChaincodeStubInterface, string) (bool, error)
    ValidateRealestate(shim.ChaincodeStubInterface, *Realestate) (bool, error)
    GetRealestate(shim.ChaincodeStubInterface, string) (*Realestate, error)
    UpdateRealestate(shim.ChaincodeStubInterface, *Realestate) error
    ListRealestates(shim.ChaincodeStubInterface) ([]*Realestate, error)

    TransferRealestate(stub shim.ChaincodeStubInterface, realestateId string, newOwnerId string)

```

저번주 지적하셨던 사항을 반영하여 최대한 간단하게 struct를 설계했습니다. 참여자를 추가하고 확인하고 리스트를 뽑아내는 함수만을 구현했습니다. 다음으로 부동산도 구분자 이름 소유자 아이디 거래 가능한지 그리고 거래 시간 정도로 설계했습니다. 함수는 부동산을 추가하고 부동산을 체크하고 부동산의 유효성을 확인하고 특정 부동산의 정보를 가져오고 모든 부동산의 정보를 가져오는 함수를 만들었습니다. 마지막으로 부동산의 거래를 통해서 부동산의 OwnerId가 바뀌는 transfer함수를 만들었습니다.

```

package chaincode_test

import (
    "sort"
    "fmt"
    "encoding/json"
    "realestatetransfer/chaincode"
    "github.com/hyperledger/fabric/common/util"
    "github.com/hyperledger/fabric/core/chaincode/shim"
    pb "github.com/hyperledger/fabric/protos/peer"
    "github.com/stretchr/testify/assert"
    "testing"
)

const (
    alice = `{"Id": "1", "Name": "Alice", "Token": "100"}`
    bob = `{"Id": "2", "Name": "Bob", "Token": "100"}`

    emptyParticipant = "[]"
    oneParticipant = "[" + alice + "]"
    twoParticipants = "[" + alice + ", " + bob + "]"

    timestamp = "2020-04-18T12:34:56Z"

    realestate1 = `{"Id": "1", "Name": "eileen-garden", "OwnerId": "1", "IsTrading": "true", "Timestamp": "2020-04-18T12:34:56Z"}`
    realestate1b = `{"Id": "1", "Name": "eileen-garden", "OwnerId": "2", "IsTrading": "true", "Timestamp": "2020-04-18T12:34:56Z"}`
    realestate2 = `{"Id": "2", "Name": "maetan-hillstate", "OwnerId": "1", "IsTrading": "true", "Timestamp": "2020-04-18T12:34:56Z"}`

    oneRealestate = "[" + realestate1 + "]"
    twoRealestates = "[" + realestate1 + ", " + realestate2 + "]"

    one = "1"
    two = "2"
)

```

테스트 파일은 impl\_test.go를 통해서 만들었으며 위와 같은 테스트 데이터를 생성해서 테스트 함수를 만들어서 테스트를 했습니다.

```

2020-04-19 21:13:04.784 KST [realstatetransfer] Info -> INFO 0c8 function name = AddParticipant
2020-04-19 21:13:04.784 KST [realstatetransfer] Info -> INFO 0c9 AddParticipant: Id = 1
2020-04-19 21:13:04.784 KST [realstatetransfer] Info -> INFO 0ca CheckParticipant: Id = 1
2020-04-19 21:13:04.784 KST [realstatetransfer] Info -> INFO 0cb Invoke called: Tx ID = e8d2414b-c60b-48d
2020-04-19 21:13:04.784 KST [realstatetransfer] Info -> INFO 0cc function name = AddRealestate
2020-04-19 21:13:04.784 KST [realstatetransfer] Info -> INFO 0cd AddRealestate: Id = 1
2020-04-19 21:13:04.784 KST [realstatetransfer] Info -> INFO 0ce CheckRealestate: Id = 1
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0cf ValidateRealestate: Id = 1
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0d0 CheckParticipant: Id = 1
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0d1 Invoke called: Tx ID = a5aea165-e806-415
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0d2 function name = TransferRealestate
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0d3 TransferRealestate: Realestate Id = 1, n
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0d4 GetRealestate: Id = 1
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0d5 UpdateRealestate: realestate = &{Id:1 Nar
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0d6 CheckRealestate: Id = 1
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0d7 ValidateRealestate: Id = 1
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0d8 CheckParticipant: Id = 2
2020-04-19 21:13:04.785 KST [realstatetransfer] Warning -> WAFN 0d9 Validate of the Realestate failed
2020-04-19 21:13:04.785 KST [realstatetransfer] Warning -> WAFN 0da Validate of the Realestate failed
--- PASS: TestTransferRealestate_NG2 (0.00s)
=== RUN TestTransferRealestate_NG3
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0db chaincode initialized
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0dc Invoke called: Tx ID = 8ec4486c-1814-4ec
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0dd function name = AddParticipant
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0de AddParticipant: Id = 1
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0df CheckParticipant: Id = 1
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0e0 Invoke called: Tx ID = b182falb-2ff8-4fc
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0e1 function name = TransferRealestate
2020-04-19 21:13:04.785 KST [realstatetransfer] Warning -> WAFN 0e2 not enough number of arguments: 1 give
--- PASS: TestTransferRealestate_NG3 (0.00s)
=== RUN TestTransferRealestate_NG4
2020-04-19 21:13:04.785 KST [realstatetransfer] Info -> INFO 0e3 chaincode initialized
2020-04-19 21:13:04.786 KST [realstatetransfer] Info -> INFO 0e4 Invoke called: Tx ID = 1a52f0a5-b195-424
2020-04-19 21:13:04.786 KST [realstatetransfer] Info -> INFO 0e5 function name = AddParticipant
2020-04-19 21:13:04.786 KST [realstatetransfer] Info -> INFO 0e6 AddParticipant: Id = 1
2020-04-19 21:13:04.786 KST [realstatetransfer] Info -> INFO 0e7 CheckParticipant: Id = 1
2020-04-19 21:13:04.786 KST [realstatetransfer] Info -> INFO 0e8 Invoke called: Tx ID = c50b9070-8ea0-43c
2020-04-19 21:13:04.786 KST [realstatetransfer] Info -> INFO 0e9 function name = TransferRealestate
2020-04-19 21:13:04.786 KST [realstatetransfer] Warning -> WAFN 0ea not enough number of arguments: 1 give
--- PASS: TestTransferRealestate_NG4 (0.00s)
=== RUN TestValidateRealestate_NG1
2020-04-19 21:13:04.786 KST [realstatetransfer] Info -> INFO 0eb chaincode initialized
2020-04-19 21:13:04.786 KST [realstatetransfer] Info -> INFO 0ec Invoke called: Tx ID = bf2cc477-56af-439
2020-04-19 21:13:04.786 KST [realstatetransfer] Info -> INFO 0ed function name = ValidateRealestate
2020-04-19 21:13:04.786 KST [realstatetransfer] Warning -> WAFN 0ee Unknown method: ValidateRealestate
--- PASS: TestValidateRealestate_NG1 (0.00s)
FAIL
coverage: 70.6% of statements
FAIL realstatetransfer/chaincode 0.162s

```

각 함수의 성공과 실패를 단순히 테스트해서 커버리지가 높지는 않지만 각 과정에서 일어날 수 있는 과정을 단순히 테스트를 했습니다.

### 3. 연구에서 미진했던 점

Go언어가 미숙하다보니 단순히 변수를 변경하고 작성하는 속성만 조금 달라졌을 뿐인데 구현이 오래 걸렸습니다. 이번 주에 과제가 없던 것이 시간을 많이 투자할 수 있어서 많은 시간을 들여서 할 수 있었다고 생각이 듭니다. 안에 구현되는 하이퍼레저 패브릭의 함수에서 리턴되는 값을 다루는 것이 조금 버겁다는 생각이 들었습니다. 아직 수준이 올라 오지 못했지만 이것도 문서와 여러 코드를 보면서 조금씩 향상시키도록 노력하겠습니다. 이번 구현을 진행하면서 개인으로 아쉬웠던 점은 Go언어에서 Map이라는 타입의 비결정성이었습니다. 처음에는 값을 넣을 때 정렬해서 넣으면 될 것 같아서 시도를 해봤습니다. 나중에 알고보니 Map에서 들어갈 때 비결정으로 키값이 들어간다는 것을 알게 되었습니다. 출력되는 결과 값에서 다시 정렬을 하는 함수를 만들어서 실행

을 해보았지만 실통하지 않았습니다. 테스트를 하는 과정에서 중요하지 않은 것 같아서 중간에 넘어갔습니다. Go언어를 깊이 공부해서 익숙하지 않은 수정이 아닌 개발로 나아갈 수 있겠음 조금 더 정진하도록 하겠습니다.

#### 4. 연구 진행시 생긴 의문점 혹은 아이디어

토큰이라는 참여자에게 새로운 값을 넣으면서 타인에게 돈을 빌려오는 과정도 있으면 어떨지 생각을 해봤습니다. 금융기관을 새로운 조직으로 넣은 다음에 채권과 채무를 지는 과정이 있다면 현실에 있는 거래 과정을 모사하는데 더 접착되지 않을까 생각을 해봤습니다. 구현은 참여자로서 개체를 생성하면 되지 않을까 생각을 해봤습니다. 테스트가 아니라 클라이언트 프로그램을 개발하는 것도 생각을 해봤습니다. 교수님이 이야기하신 Multsig를 테스트를 하는 방식이 아닌 웹이나 앱으로 보여서 실제감을 느끼게 하는 것을 어떨지 생각을 해봤습니다.