Chaincode development

deploy network

```
# start network
./network.sh up createChannel -ca -s couchdb

# deploy chaincode
./network.sh deployCC -ccn hyperForex -ccv 1.0 -ccp /home/ubuntu/hyperForeX/chaincode -ccl javascript

# create users
cd ~/hyperForeX/chaincode
npm i
node bin/enrollUser.js 'CAAdmin@org1.example.com' admin adminpw
node bin/enrollUser.js 'CAAdmin@org2.example.com' admin adminpw

Then register user as follows:

node bin/registerUser.js 'CAAdmin@org2.example.com' 'Alice@org2.example.com' '{"secret": "userpw"}'
node bin/enrollUser.js 'Alice@org2.example.com' 'Alice@org2.example.com' userpw

node bin/registerUser.js 'CAAdmin@org2.example.com' 'Bob@org2.example.com' '{"secret": "userpw"}'
node bin/enrollUser.js 'Bob@org2.example.com' 'Bob@org2.example.com' userpw
```

```
+ peer lifecycle chaincode checkcommitreadiness --channelID mychannel --name hyperForex --version 1.0 --sequence 1 --output json
+ res=0
{
        "approvals": {
                "Org1MSP": true,
                "Org2MSP": true
        }
}
Checking the commit readiness of the chaincode definition successful on peer0.org2 on channel 'mychannel'
Using organization 1
Using organization 2
+ peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/ubuntu/fabric-samples
ert.pem --channelID mychannel --name hyperForex --peerAddresses localhost:7051 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/orgo
eerAddresses localhost:9051 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/tlsca/
+ res=0
2023-06-21 22:35:38.296 UTC 0001 INFO [chaincodeCmd] ClientWait -> txid [53523487fe7c6949f157bdef06136f287578acebcc92e2935f46400660d39c54] comm
2023-06-21 22:35:38.298 UTC 0002 INFO [chaincodeCmd] ClientWait -> txid [53523487fe7c6949f157bdef06136f287578acebcc92e2935f46400660d39c54] comm
Chaincode definition committed on channel 'mychannel'
Using organization 1
Querying chaincode definition on peer0.org1 on channel 'mychannel'...
Attempting to Query committed status on peer0.org1, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name hyperForex
+ res=0
Committed chaincode definition for chaincode 'hyperForex' on channel 'mychannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org1 on channel 'mychannel'
Using organization 2
Querying chaincode definition on peer0.org2 on channel 'mychannel'...
Attempting to Query committed status on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name hyperForex
+ res=0
Committed chaincode definition for chaincode 'hyperForex' on channel 'mychannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org2 on channel 'mychannel'
Chaincode initialization is not required
ubuntu@ip-172-31-87-211:~/fabric-samples/test-network$
```

```
found 0 vulnerabilities
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/enrollUser.js 'CAAdmin@org1.example.com' admin adminpw
Successfully enrolled CAAdmin@org1.example.com user and imported it into the wallet
User enrollment completed successfully.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/enrollUser.js 'CAAdmin@org2.example.com' admin adminpw
Successfully enrolled CAAdmin@org2.example.com user and imported it into the wallet
User enrollment completed successfully.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/registerUser.js 'CAAdmin@org2.example.com' 'Alice@org2.example.com' '{"secret": "userpw"}'
Successfully registered the user with the Alice@org2.example.com enrollment ID and userpw enrollment secret.
User registration completed successfully.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/enrollUser.js 'Alice@org2.example.com' 'Alice@org2.example.com' userpw
Successfully enrolled Alice@org2.example.com user and imported it into the wallet
User enrollment completed successfully.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/registerUser.js 'CAAdmin@org2.example.com' 'Bob@org2.example.com' '{"secret": "userpw"}'
Successfully registered the user with the Bob@org2.example.com enrollment ID and userpw enrollment secret.
User registration completed successfully.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/enrollUser.js 'Bob@org2.example.com' 'Bob@org2.example.com' userpw
Successfully enrolled Bob@org2.example.com user and imported it into the wallet
User enrollment completed successfully.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$
```

Test by submitTransaction.js

add currency:

```
node bin/submitTransaction.js 'CAAdmin@org1.example.com' currencyList
node bin/submitTransaction.js 'CAAdmin@org1.example.com' addCurrency USD 1.0
node bin/submitTransaction.js 'CAAdmin@org1.example.com' addCurrency CAD 0.75
node bin/submitTransaction.js 'CAAdmin@org1.example.com' currencyList
```

```
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'CAAdmin@org1.example.com' currencyList
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit currencyList transaction.
Response from currencyList: []
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'CAAdmin@org1.example.com' addCurrency USD 1.0
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit addCurrency transaction.
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'CAAdmin@org1.example.com' addCurrency CAD 0.75
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit addCurrency transaction.
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'CAAdmin@org1.example.com' currencyList
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit currencyList transaction.
Response from currencyList: [{"currency":"USD","exchangeRateToUSD":1},{"currency":"CAD","exchangeRateToUSD":0.75}]
Disconnect from the gateway.
```

update rate :

```
node bin/submitTransaction.js 'CAAdmin@org1.example.com' updateRate CAD 0.7
node bin/submitTransaction.js 'CAAdmin@org1.example.com' currencyList
```

```
Submit currencyList transaction.
Response from currencyList: [{"currency":"USD","exchangeRateToUSD":1},{"currency":"CAD","exchangeRateToUSD":0.75}]
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'CAAdmin@org1.example.com' updateRate CAD 0.7
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit updateRate transaction.
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'CAAdmin@org1.example.com' currencyList
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit currencyList transaction.
Response from currencyList: [{"currency":"USD","exchangeRateToUSD":1},{"currency":"CAD","exchangeRateToUSD":0.7}]
Disconnect from the gateway.
```

init account, check balance:

```
node bin/submitTransaction.js 'CAAdmin@org1.example.com' initAccount 'CAAdmin@org1.example.com'

node bin/submitTransaction.js 'CAAdmin@org1.example.com' balance 'CAAdmin@org1.example.com'
```

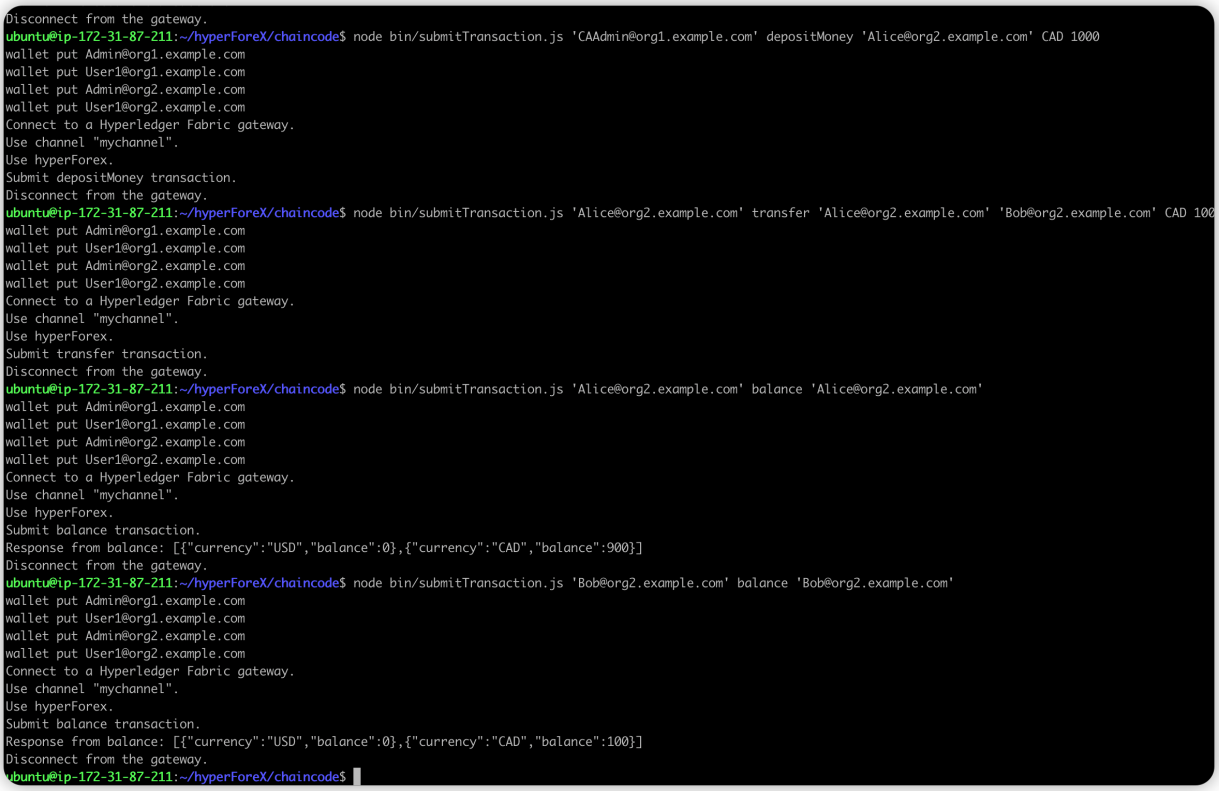
```
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'CAAdmin@org1.example.com' initAccount 'CAAdmin@org1.example.com'
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit initAccount transaction.
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'CAAdmin@org1.example.com' balance 'CAAdmin@org1.example.com'
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit balance transaction.
Response from balance: [{"currency":"USD","balance":0},{"currency":"CAD","balance":0}]
Disconnect from the gateway.
```

deposit and withdraw:

```
node bin/submitTransaction.js 'CAAdmin@org1.example.com' depositMoney 'CAAdmin@org1.example.com' USD 100

node bin/submitTransaction.js 'CAAdmin@org1.example.com' balance 'CAAdmin@org1.example.com'

node bin/submitTransaction.js 'CAAdmin@org1.example.com' withdrawMoney 'CAAdmin@org1.example.com' USD 50

node bin/submitTransaction.js 'CAAdmin@org1.example.com' balance 'CAAdmin@org1.example.com'
```

transfer:

```
# init Alice & Bob account:
node bin/submitTransaction.js 'Alice@org2.example.com' initAccount 'Alice@org2.example.com'
node bin/submitTransaction.js 'Bob@org2.example.com' initAccount 'Bob@org2.example.com'
# deposit money:
node bin/submitTransaction.js 'CAAdmin@org1.example.com' depositMoney 'Alice@org2.example.com' CAD 1000
# transfer:
node bin/submitTransaction.js 'Alice@org2.example.com' transfer 'Alice@org2.example.com' 'Bob@org2.example.com' CAD 100
node bin/submitTransaction.js 'Alice@org2.example.com' balance 'Alice@org2.example.com'
node bin/submitTransaction.js 'Bob@org2.example.com' balance 'Bob@org2.example.com'
```



exchange money:

```
node bin/submitTransaction.js 'Alice@org2.example.com' exchangeCurrency 'Alice@org2.example.com' CAD USD 500 0
node bin/submitTransaction.js 'Alice@org2.example.com' balance 'Alice@org2.example.com'

node bin/submitTransaction.js 'Alice@org2.example.com' exchangeCurrency 'Alice@org2.example.com' USD CAD 0 200
node bin/submitTransaction.js 'Alice@org2.example.com' balance 'Alice@org2.example.com'
```

```
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'Alice@org2.example.com' exchangeCurrency 'Alice@org2.example.com' CAD USD 500 0
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit exchangeCurrency transaction.
Response from exchangeCurrency: {"amountFrom":500,"amountTo":350,"balanceOfCurrencyFrom":400,"balanceOfCurrencyTO":350}
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'Alice@org2.example.com' balance 'Alice@org2.example.com'
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit balance transaction.
Response from balance: [{"currency":"USD","balance":350},{"currency":"CAD","balance":400}]
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'Alice@org2.example.com' exchangeCurrency 'Alice@org2.example.com' USD CAD 0 200
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit exchangeCurrency transaction.
Response from exchangeCurrency: {"amountFrom":140,"amountTo":200,"balanceOfCurrencyFrom":210,"balanceOfCurrencyTO":600}
Disconnect from the gateway.
ubuntu@ip-172-31-87-211:~/hyperForeX/chaincode$ node bin/submitTransaction.js 'Alice@org2.example.com' balance 'Alice@org2.example.com'
wallet put Admin@org1.example.com
wallet put User1@org1.example.com
wallet put Admin@org2.example.com
wallet put User1@org2.example.com
Connect to a Hyperledger Fabric gateway.
Use channel "mychannel".
Use hyperForex.
Submit balance transaction.
Response from balance: [{"currency":"USD","balance":210},{"currency":"CAD","balance":600}]
Disconnect from the gateway.
```