Contents

[Project build: 3](#_Toc181395591)

[Chaincode setup: 3](#_Toc181395592)

[Setting up the Hyperledger Fabric Test network 4](#_Toc181395593)

[Create Channel : mychannel 4](#_Toc181395594)

[Package: 4](#_Toc181395595)

[Install: 5](#_Toc181395596)

[Approve: 6](#_Toc181395597)

[Commit: 6](#_Toc181395598)

[Invoke 7](#_Toc181395599)

[Initial method test: 7](#_Toc181395600)

[Created patient 7](#_Toc181395601)

[Method name : createMedicalRecord 7](#_Toc181395602)

[Arguments : String patientId, String name, String diagnosis,String treatment, String dateofVisit 7](#_Toc181395603)

[Command: 7](#_Toc181395604)

[Result: 7](#_Toc181395605)

[View patient record 8](#_Toc181395606)

[Method name: viewMedicalRecord 8](#_Toc181395607)

[Arguments: String patientId 8](#_Toc181395608)

[Command: 8](#_Toc181395609)

[Result 8](#_Toc181395610)

[Not exist patient 8](#_Toc181395611)

[Same patient creation 8](#_Toc181395612)

[Update record 8](#_Toc181395613)

[Method name: updateMedicalRecord 8](#_Toc181395614)

[Arguments : String patientId, String diagnosis, String treatment,String dateOfVisit 8](#_Toc181395615)

[Command: 8](#_Toc181395616)

[Result: 9](#_Toc181395617)

[Query updated details: 9](#_Toc181395618)

[Create reports: 9](#_Toc181395619)

[Method name: addTestReport 9](#_Toc181395620)

[Arguments: String patientId, String reportId, String reportName, String description, String 9](#_Toc181395621)

[Command: 9](#_Toc181395622)

[Result: 9](#_Toc181395623)

[Query reports: 10](#_Toc181395624)

[Create same report: 10](#_Toc181395625)

[Create another report: 10](#_Toc181395626)

[Query test report: 10](#_Toc181395627)

[Update test report 11](#_Toc181395628)

[Method name: updateTestReport 11](#_Toc181395629)

[Argument: String patientId, String reportId, String reportName, String description, String reportDate, String resportIssuer 11](#_Toc181395630)

[Command: 11](#_Toc181395631)

[Result: 11](#_Toc181395632)

[Add medicine: 12](#_Toc181395633)

[Method name: addMedicine 12](#_Toc181395634)

[Arguments: String patientId, String medicineName, String medicineExpiry, String medicineTakenDate 12](#_Toc181395635)

[Command: 12](#_Toc181395636)

[Result: 12](#_Toc181395637)

[Add more medicine: 12](#_Toc181395638)

[Query medicine taken on particular day: 12](#_Toc181395639)

[Method name: viewMedicineTakenOnParticularDate 12](#_Toc181395640)

[Arguments: String patientId, String date 12](#_Toc181395641)

[Command: 12](#_Toc181395642)

[Result: 13](#_Toc181395643)

[Query medicine taken between days 13](#_Toc181395644)

[Method name: viewMedicineTakenOnStartAndEndDate 13](#_Toc181395645)

[Arguments: String patientId, String startDate, String endDate 13](#_Toc181395646)

[Command: 13](#_Toc181395647)

[Result: 13](#_Toc181395648)

[Get all medicine: 13](#_Toc181395649)

[Method name: getAllMedicine 13](#_Toc181395650)

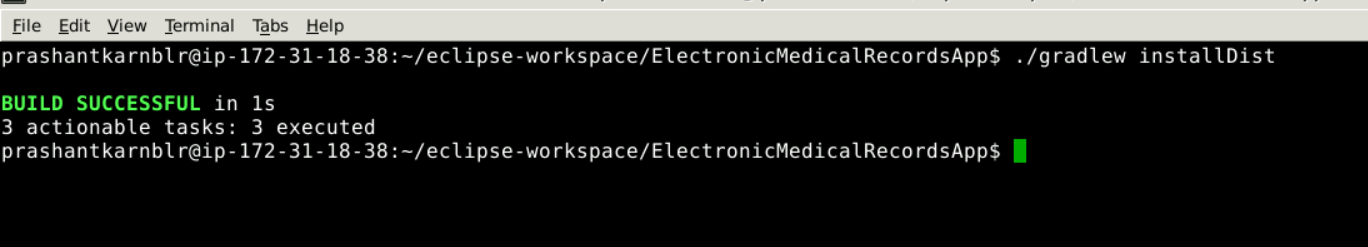
[Arguments: String patientId 13](#_Toc181395651)

[Command: 13](#_Toc181395652)

[Result: 14](#_Toc181395653)

# Project build:

**./gradlew installDist**



# Chaincode setup:

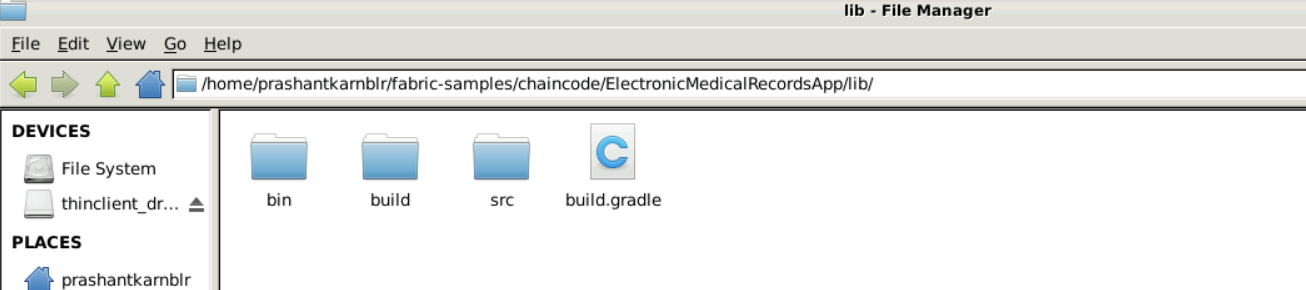
Paste Project inside **chaincode** folder of fabric-samples



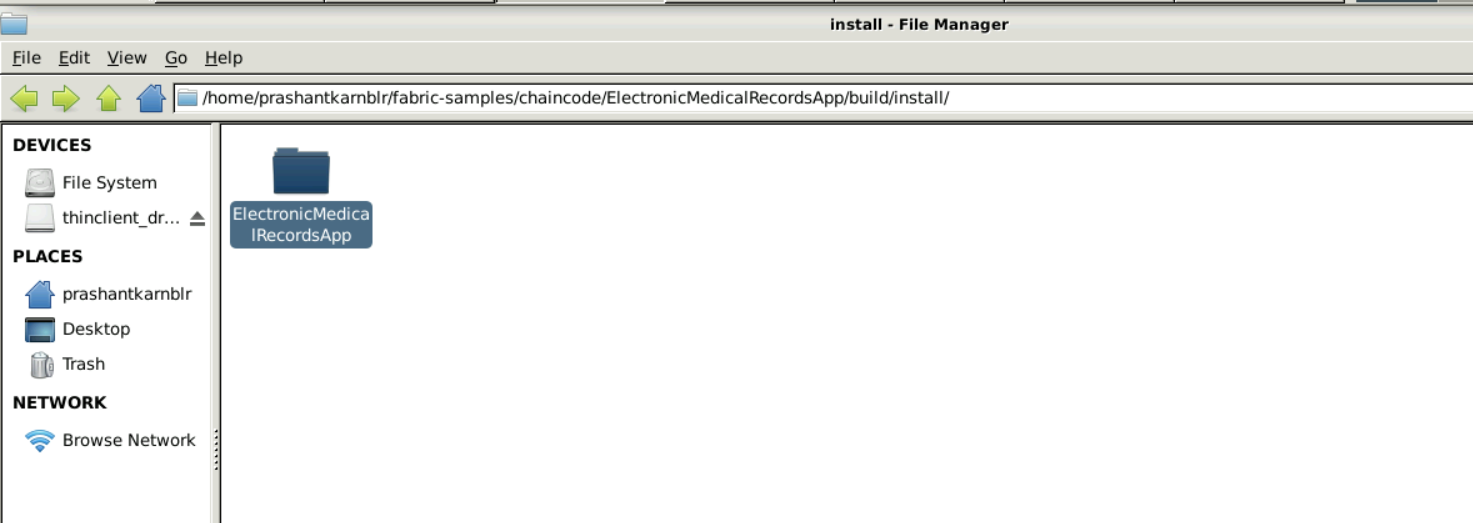




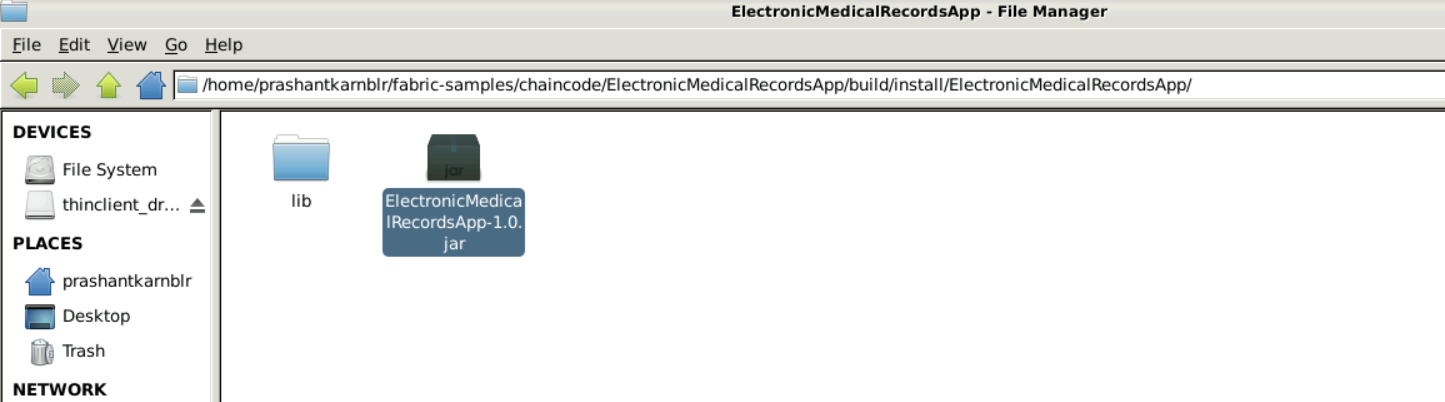
Go inside <project>/lib folder and copy all four files and paste into inside <project>



Change lib folder name to <project> name.



Change lib-1.0.jar to <project>-1.0.jar



# Setting up the Hyperledger Fabric Test network

1. Navigate to fabric-samples/test-networks:

***cd fabric-samples/test-network***

1. Stop the previously running test network by running the following command:

***sudo ./network.sh down***

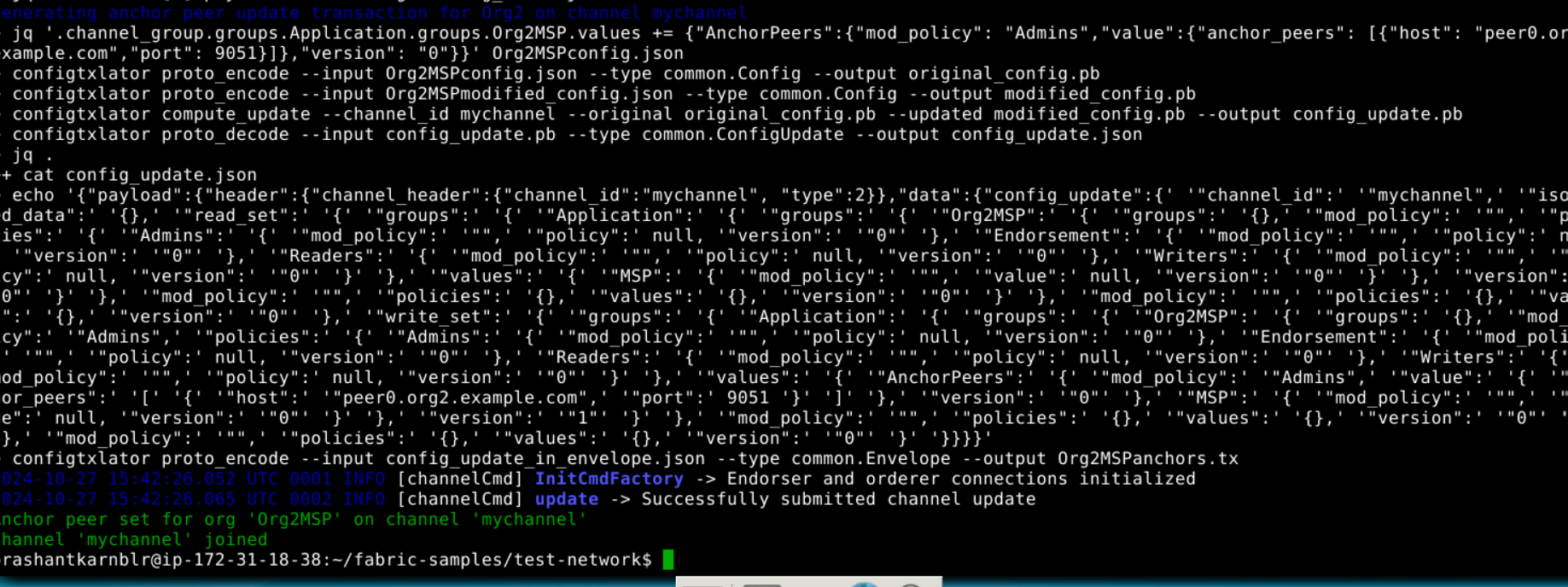
1. Remove the unused docker images by running the following command:

***sudo docker system prune***

1. Start the test network by executing the following command:

***sudo ./network.sh up -ca -s couchdb***

# Create Channel : mychannel



# Package:

*nano lifecycle\_setup\_org1.sh*

**Add this script:**

***#!/bin/sh***

***export PATH=${PWD}/../bin:${PWD}:$PATH***

***export FABRIC\_CFG\_PATH=$PWD/../config/***

***export CORE\_PEER\_TLS\_ENABLED=true***

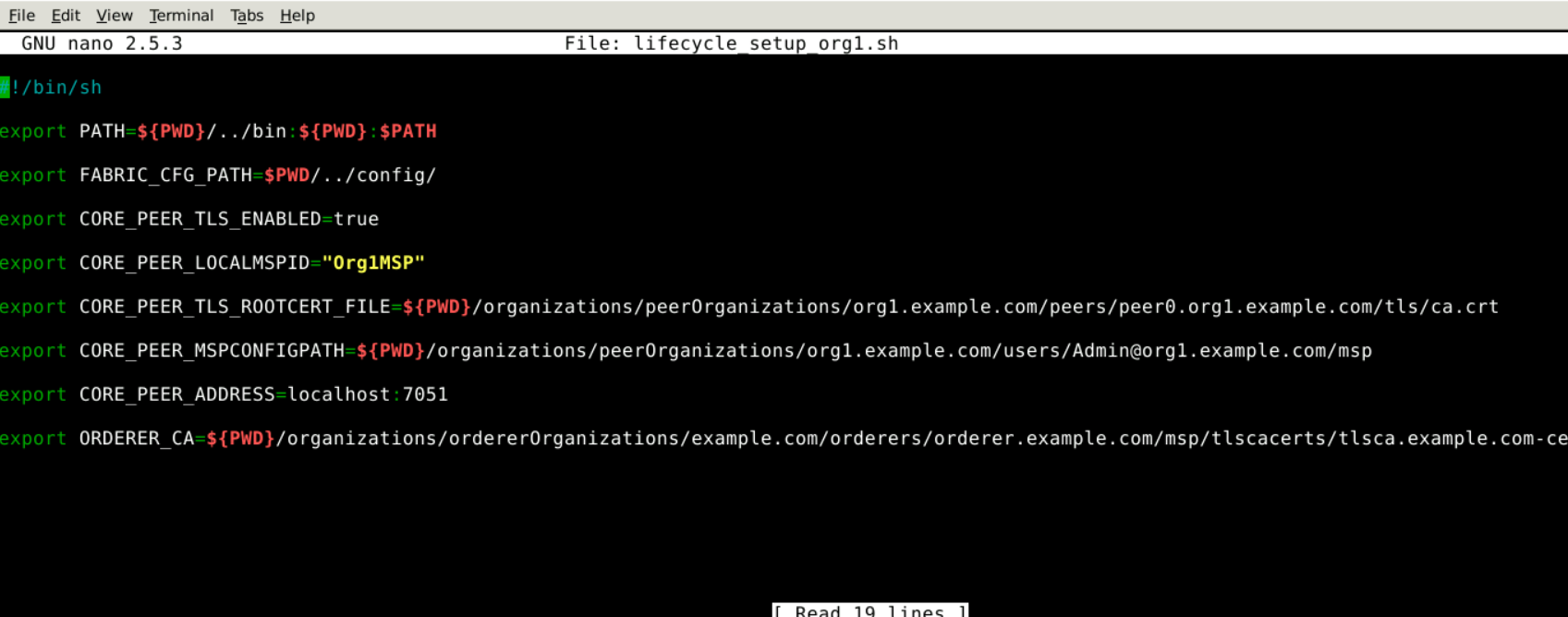
***export CORE\_PEER\_LOCALMSPID="Org1MSP"***

***export CORE\_PEER\_TLS\_ROOTCERT\_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt***

***export CORE\_PEER\_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp***

***export CORE\_PEER\_ADDRESS=localhost:7051***

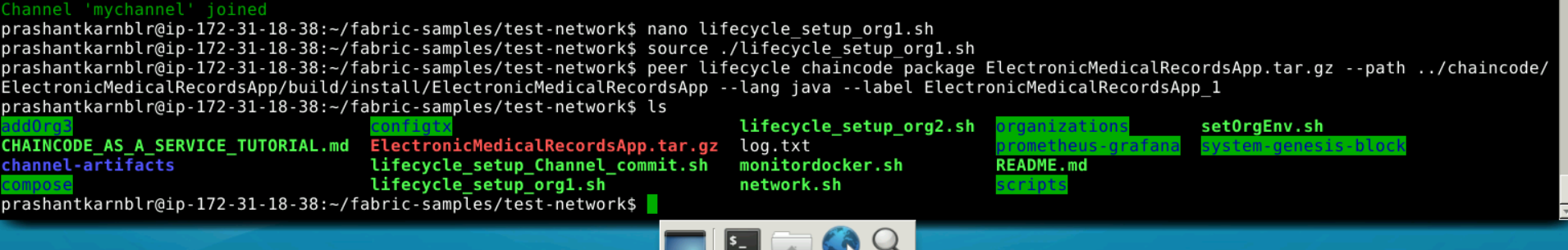
***export ORDERER\_CA=${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem***



Ctrl + X 🡪press Y

source ./lifecycle\_setup\_org1.sh

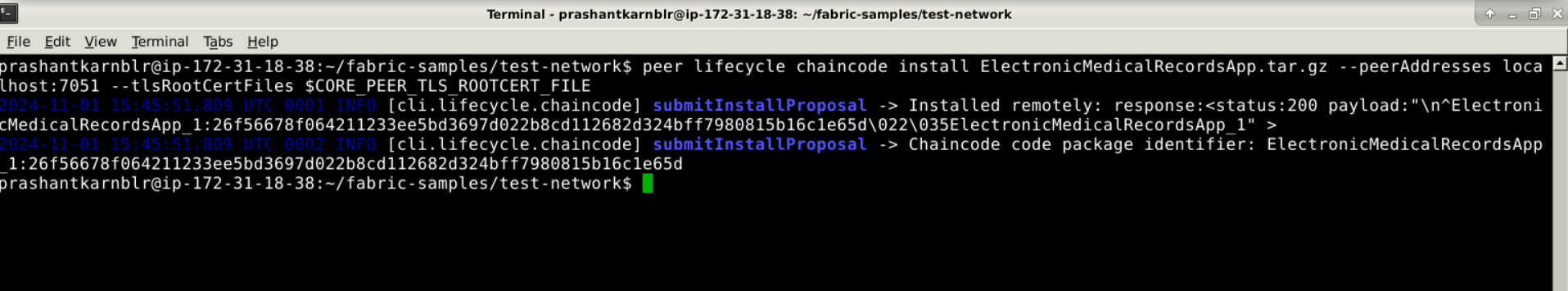
peer lifecycle chaincode package ElectronicMedicalRecordsApp.tar.gz --path ../chaincode/ElectronicMedicalRecordsApp/build/install/ElectronicMedicalRecordsApp --lang java --label ElectronicMedicalRecordsApp\_1



# Install:

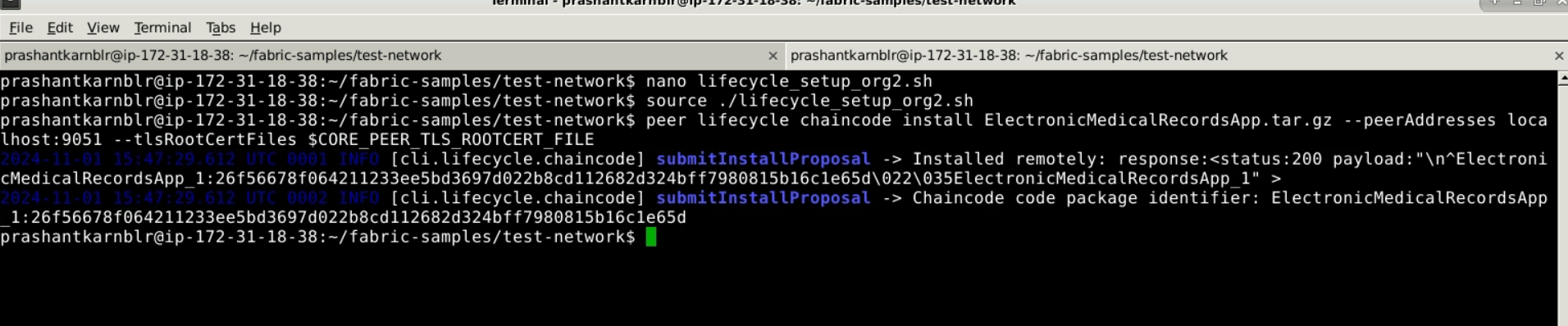
For organization 1:

peer lifecycle chaincode install ElectronicMedicalRecordsApp.tar.gz --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE



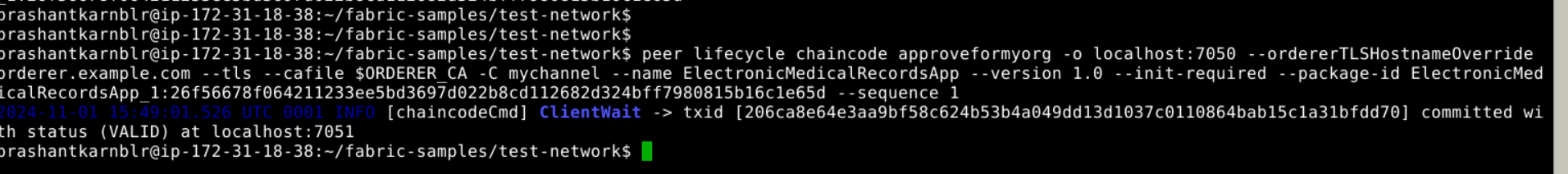
For org2:

peer lifecycle chaincode install ElectronicMedicalRecordsApp.tar.gz --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE

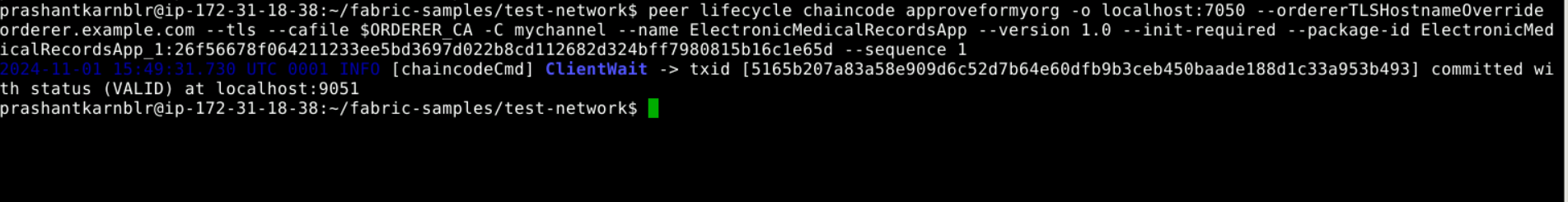


# Approve:

peer lifecycle chaincode approveformyorg -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $ORDERER\_CA -C mychannel --name ElectronicMedicalRecordsApp --version 1.0 --init-required --package-id ElectronicMedicalRecordsApp\_1:26f56678f064211233ee5bd3697d022b8cd112682d324bff7980815b16c1e65d --sequence 1



For org2:



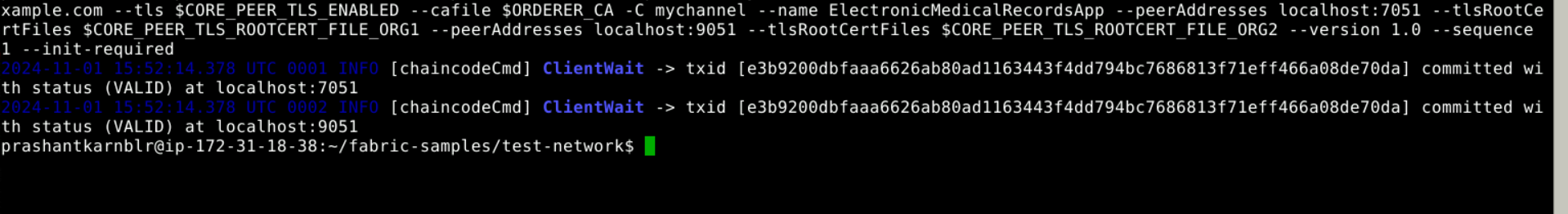
# Commit:

*nano lifecycle\_setup\_Channel\_commit.sh*

*source ./lifecycle\_setup\_Channel\_commit.sh*

peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel --name ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 --version 1.0 --sequence 1 --init-required

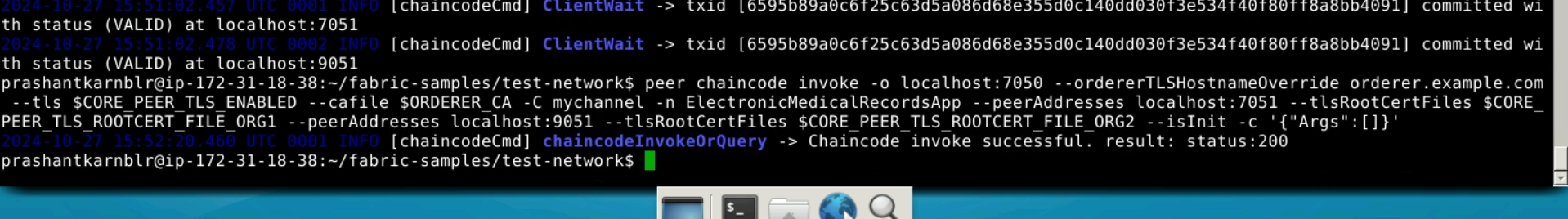
For org1:



# Invoke

## Initial method test:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 --isInit -c '{"Args":[]}'



## Created patient

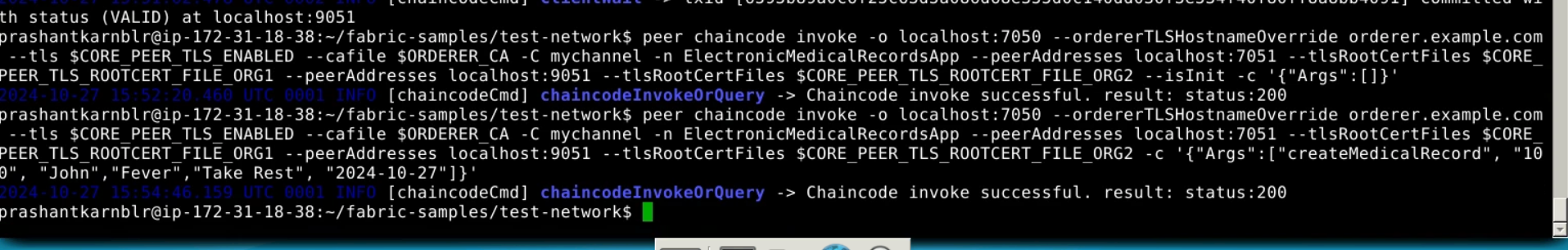
### Method name : createMedicalRecord

### Arguments : String patientId, String name, String diagnosis,String treatment, String dateofVisit

### Command:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["createMedicalRecord", "100", "John","Fever","Take Rest", "2024-10-27"]}'

### Result:



## View patient record

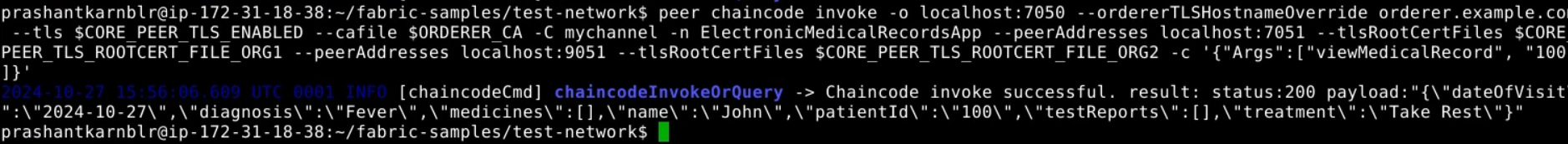
### Method name: viewMedicalRecord

### Arguments: String patientId

### Command:

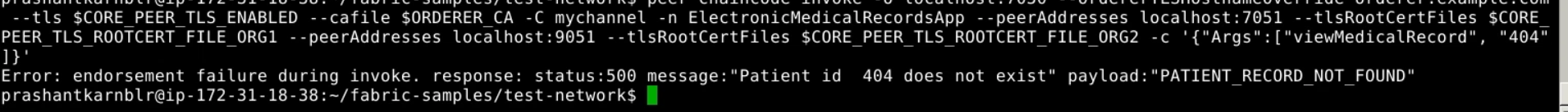
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["viewMedicalRecord", "100"]}'

### Result



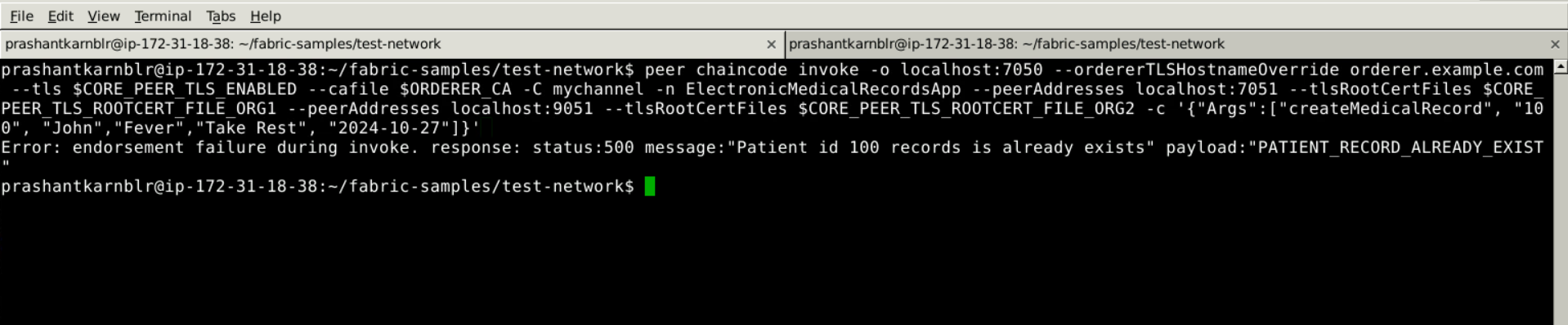
### Not exist patient

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["viewMedicalRecord", "404"]}'



### Same patient creation

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["createMedicalRecord", "100", "John","Fever","Take Rest", "2024-10-27"]}'



## Update record

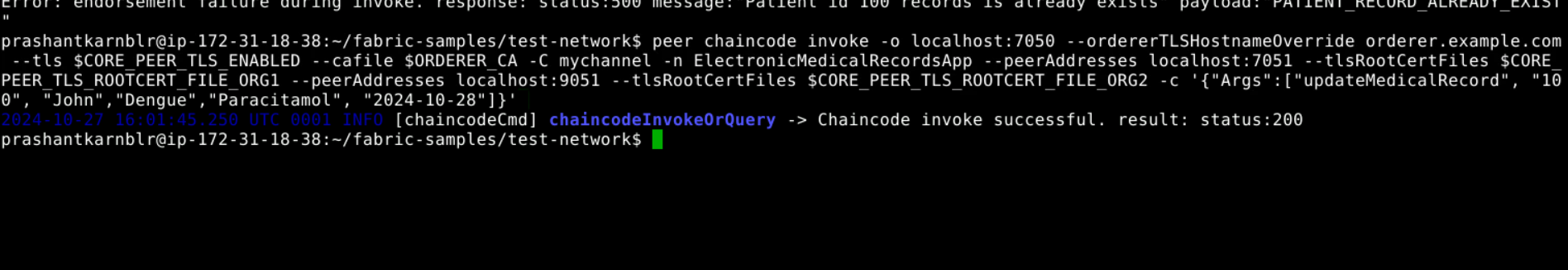
### Method name: updateMedicalRecord

### Arguments : String patientId, String diagnosis, String treatment,String dateOfVisit

### Command:

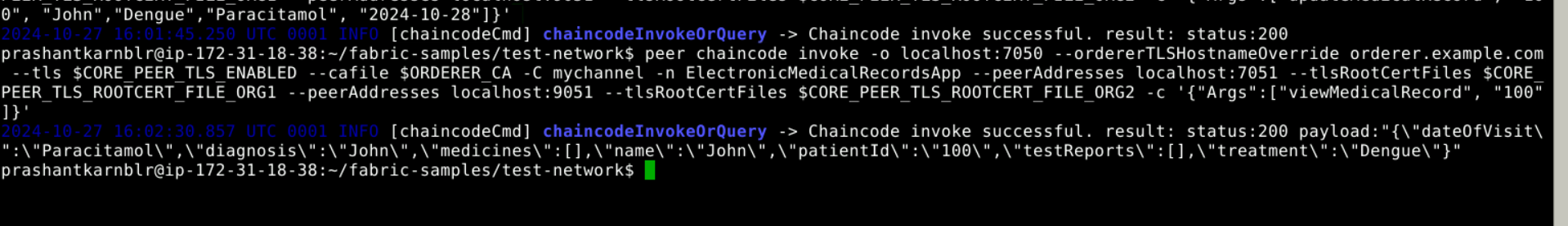
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["updateMedicalRecord", "100", "John","Dengue","Paracitamol", "2024-10-28"]}'

### Result:



### Query updated details:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["viewMedicalRecord", "100"]}'



## Create reports:

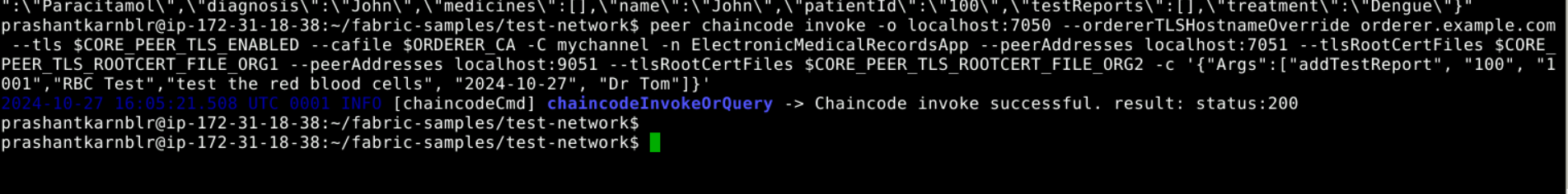
### Method name: addTestReport

Arguments: String patientId, String reportId, String reportName, String description, String reportDate, String resportIssuer

### Command:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["addTestReport", "100", "1001","RBC Test","test the red blood cells", "2024-10-27", "Dr Tom"]}'

### Result:



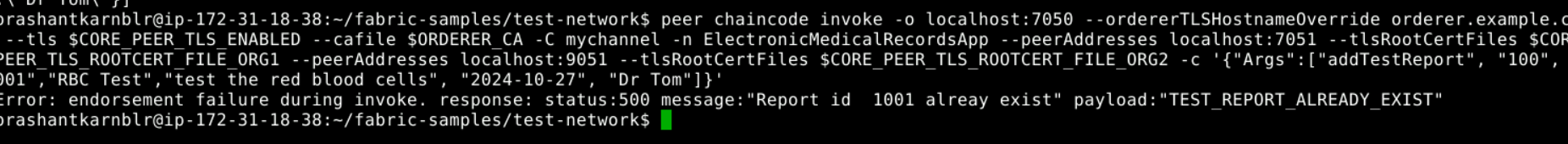
### Query reports:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["viewTestReports", "100"]}'



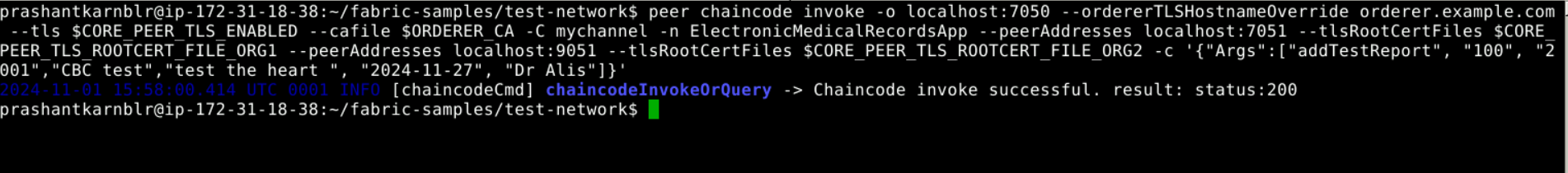
### Create same report:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["addTestReport", "100", "1001","RBC Test","test the red blood cells", "2024-10-27", "Dr Tom"]}'



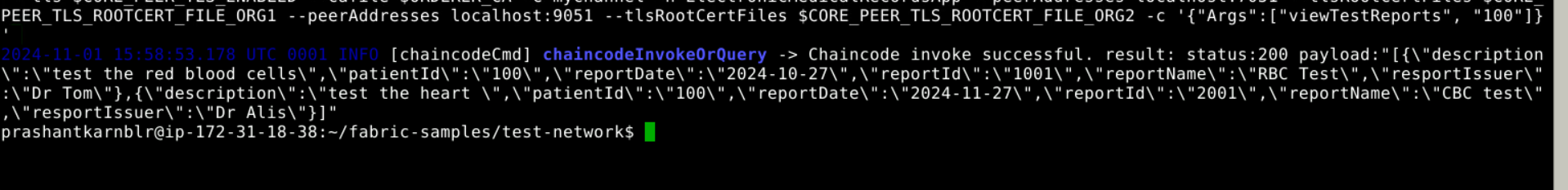
### Create another report:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["addTestReport", "100", "2001","CBC test","test the heart ", "2024-11-27", "Dr Alis"]}'



### Query test report:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["viewTestReports", "100"]}'



## Update test report

### Method name: updateTestReport

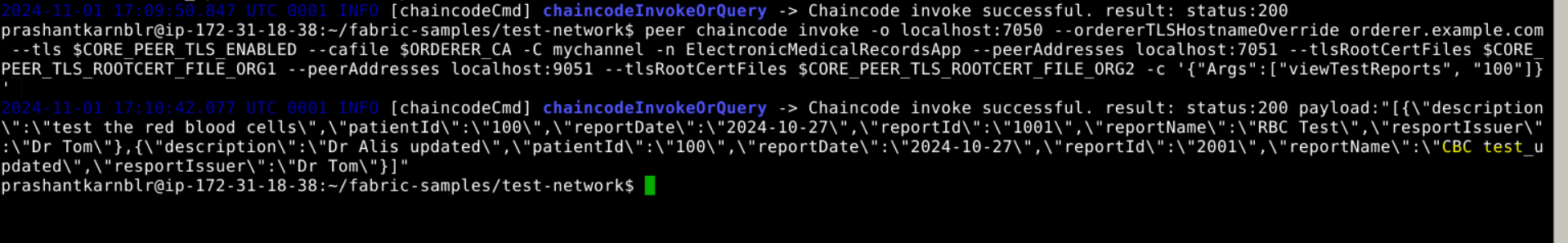
### Argument: String patientId, String reportId, String reportName, String description, String reportDate, String resportIssuer

### Command:

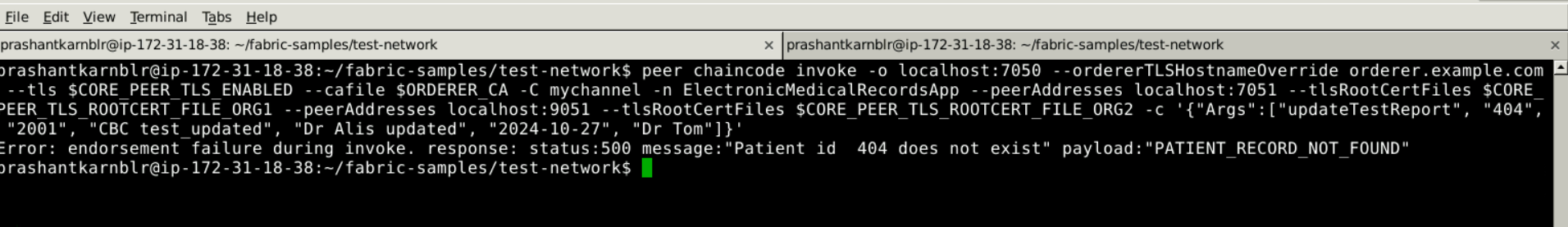
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["updateTestReport", "100", "2001", "CBC test\_updated", "Dr Alis updated", "2024-10-27", "Dr Tom"]}'

### Result:

query test report after update

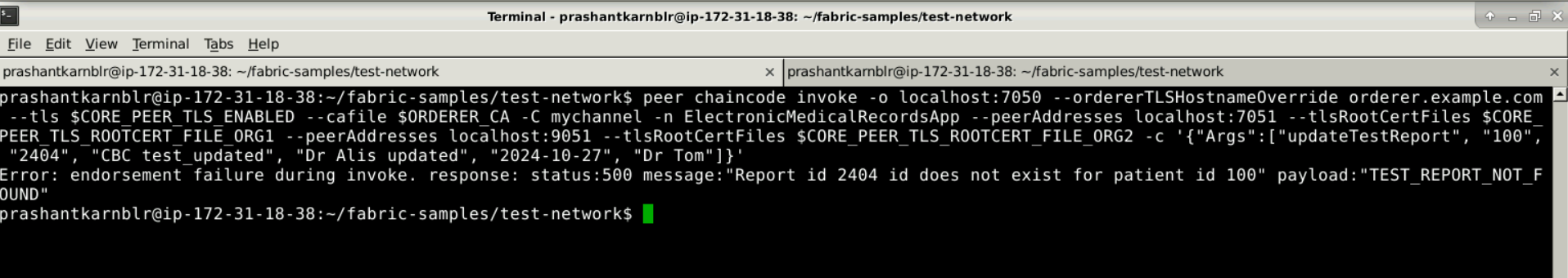


Trying to update the test report for patient which doesn’t exist:



Tyring to update test report which doesn’t exist for patient

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["updateTestReport", "100", "2404", "CBC test\_updated", "Dr Alis updated", "2024-10-27", "Dr Tom"]}'



## Add medicine:

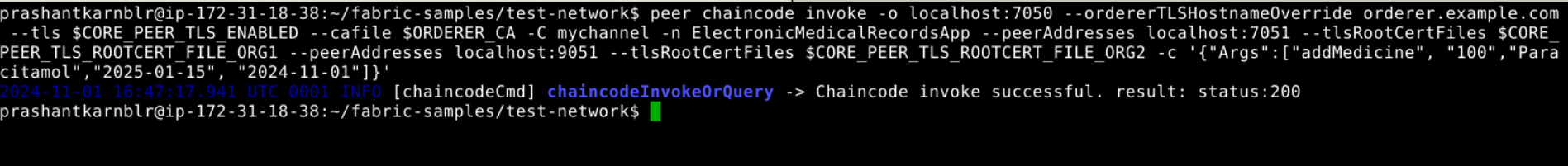
### Method name: addMedicine

### Arguments: String patientId, String medicineName, String medicineExpiry, String medicineTakenDate

### Command:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["addMedicine", "100","Paracitamol","2025-01-15", "2024-11-01"]}'

### Result:



### Add more medicine:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["addMedicine", "100","Paracitamol","2025-01-15", "2024-11-02"]}'



## Query medicine taken on particular day:

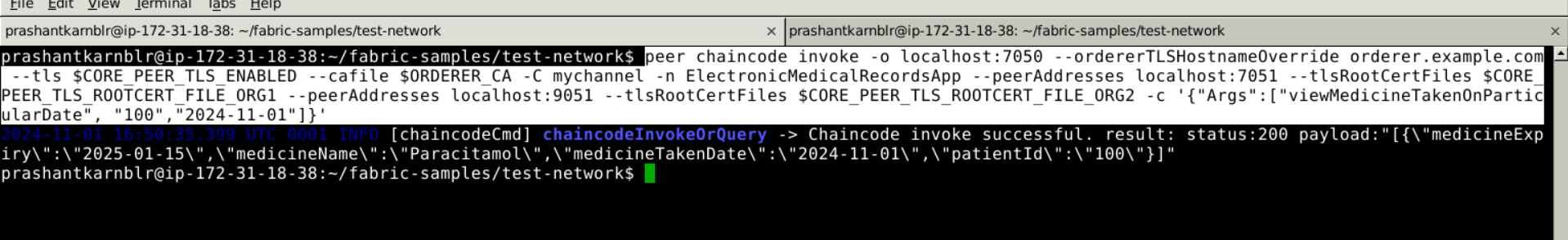
### Method name: viewMedicineTakenOnParticularDate

### Arguments: String patientId, String date

### Command:

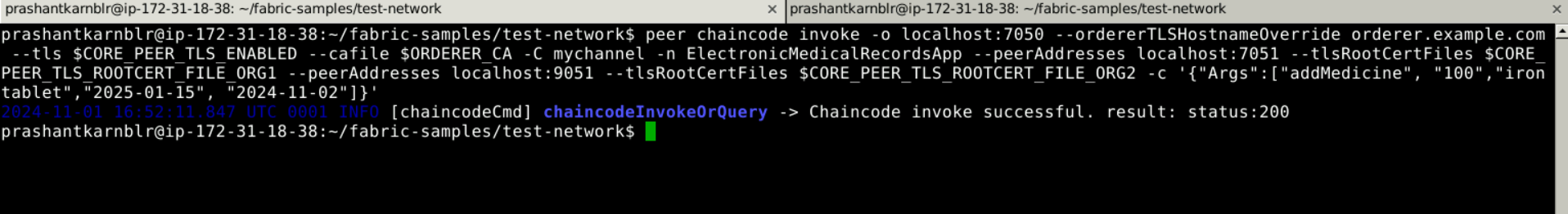
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["viewMedicineTakenOnParticularDate", "100","2024-11-01"]}'

### Result:



Adding more medicine:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["addMedicine", "100","irontablet","2025-01-15", "2024-11-02"]}'



## Query medicine taken between days

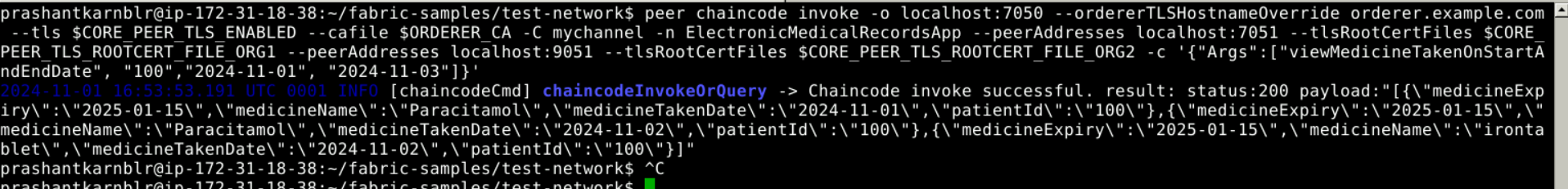
### Method name: viewMedicineTakenOnStartAndEndDate

### Arguments: String patientId, String startDate, String endDate

### Command:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["viewMedicineTakenOnStartAndEndDate", "100","2024-11-01", "2024-11-03"]}'

### Result:



## Get all medicine:

### Method name: getAllMedicine

### Arguments: String patientId

### Command:

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls $CORE\_PEER\_TLS\_ENABLED --cafile $ORDERER\_CA -C mychannel -n ElectronicMedicalRecordsApp --peerAddresses localhost:7051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG1 --peerAddresses localhost:9051 --tlsRootCertFiles $CORE\_PEER\_TLS\_ROOTCERT\_FILE\_ORG2 -c '{"Args":["getAllMedicine", "100"]}'

### Result:

