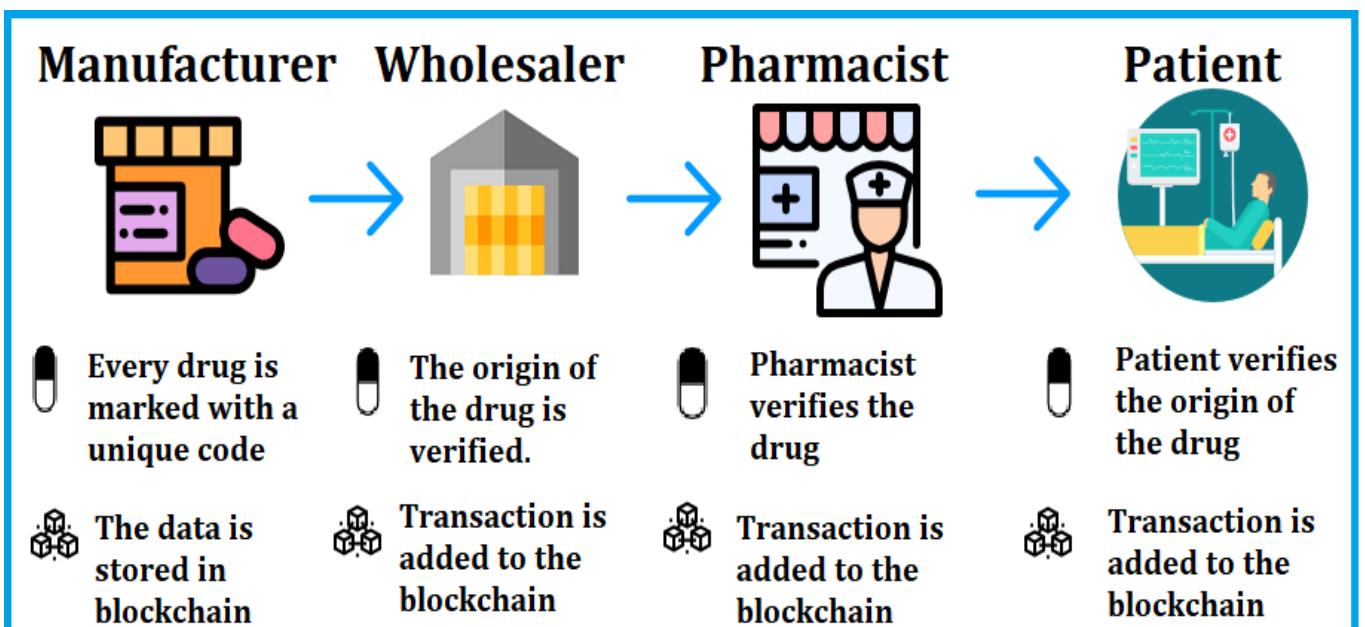


HYPER LEDGER



By

Manikandan Ramesh

PGD- Blockchain Developement

Contents

1. Introduction	5
1.1. Network Stakeholders:.....	5
2. Network Topology.....	6
2.1. Orderer Properties.....	6
2.2. Organization Peers and CA's.....	6
3. Project Structure	7
3.1. network.....	8
3.2. chaincode.....	8
3.3. application	10
3.4. test 10	
4. Pre setup steps.....	11
5. Bootstrap pharma-network.....	1
6. Install and Instantiate Chaincode.....	5
6.1. Install and Instantiate pharmanet Chaincode.....	5
6.2. Verify chaincode instantiation	7
7. Start Node Application.....	9
7.1. Install Dependencies	9
7.2. Start Node Application.....	10
8. Testing Through Postman Collection Files	11
8.1. Import Postman Collection Files	11
8.2. Initiation Test Cases	13
8.2.1. SUN Pharma	14
8.2.2. FedEx	17
8.2.3. Blue Dart	19

8.2.4. VG Pharma	21
8.2.5. upgrad	23
8.2.6. Paracetamol-001 to Paracetamol-004	26
8.2.6.1. <i>Paracetamol-001</i>	28
8.2.6.2. <i>Paracetamol-002</i>	29
8.2.6.3. <i>Paracetamol-003</i>	30
8.2.6.4. <i>Paracetamol-004</i>	31
8.3. Supply Chain Test Cases	32
8.3.1. Part a	1
8.3.1.1. <i>VGPharma-PO</i>	4
8.3.1.2. <i>Sun Pharma-Shipment</i>	5
8.3.1.3. <i>FedEx-Deliver Shipment</i>	7
8.3.2. Part b	11
8.3.2.1. <i>upgrad-PO</i>	14
8.3.2.2. <i>VG Pharma-Shipment</i>	1
8.3.1.4. <i>Blue Dart-DeliverShipment</i>	2
8.3.3. Part c	4
8.3.3.1. <i>upgrad-Retail Drug</i>	7
8.4. History Test Cases	8
8.4.1. <i>ViewDrugState</i>	9
8.4.2. <i>CheckDrugHistory</i>	12
8.5. View Wallets	17

1. INTRODUCTION

Drug counterfeiting is a major problem faced by pharma supply chain eco-systems worldwide and has potential negative consequences alike for all stakeholders. This Capstone project is intended to build a Blockchain solution using Hyperledger Fabric to tackle the drug sham challenge through tamper-evident architecture provided by Blockchain technology. Current document explains the overall network architecture, various stakeholders modeled, process of starting the network, installing and instantiating chain code and testing out various transactions.

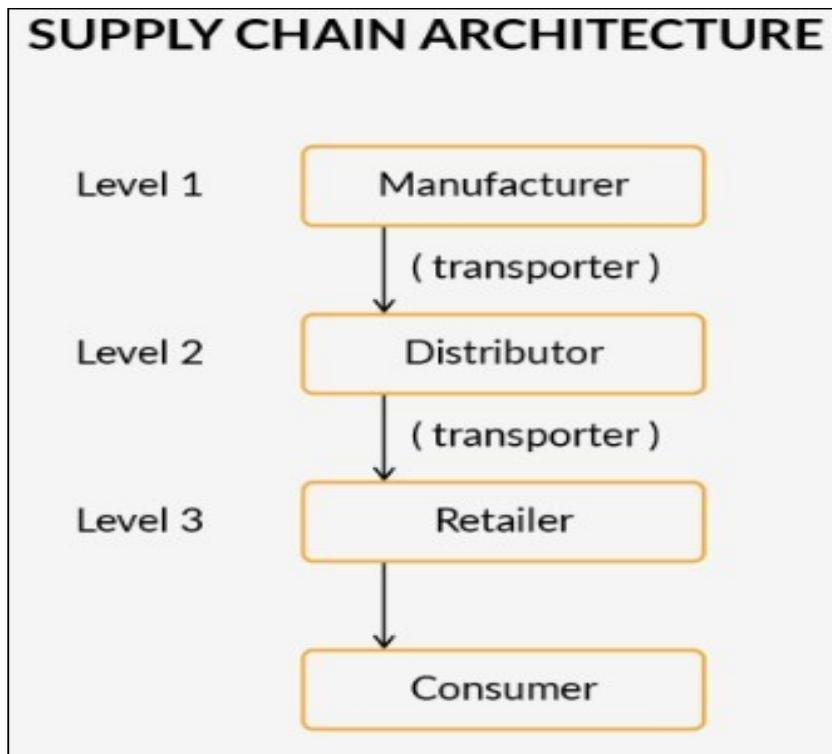
1.1. NETWORK STAKEHOLDERS:

There are five stakeholder organizations in the supply chain network -

- Manufacturer
- Distributor
- Retailer
- Consumer
- Transporter

Each of these orgs contributes two peers to the Hyperledger fabric network that is setup.

Interaction workflow between these orgs is depicted below-



2. NETWORK TOPOLOGY

- Name of the network is **pharma-network**
- There is one orderer on the network, 2 peers for each of the 5 orgs for a total of 10 peers, 5 certification authorities one per org and a CLI node - amounting to **17 docker containers** that are started as part of network setup.
- Each organization has only “Admin” user
- TLS is disabled on the network

2.1. ORDERER PROPERTIES

- Orderer Type** : Solo
- Orderer Name** : orderer.pharma-network.com

2.2. ORGANIZATION PEERS AND CA'S

Each of the stakeholder organization has -

- One Certificate Authority
- Two Peers

The properties of each peer are demonstrated in below table -

Organization	CA Name	Peer Name	Anchor Peer	Chaincode Installed	Endorser On Network	Endorser For Client App
Manufacturer	ca.manufacturer.pharma-network.com	peer0.manufacturer.pharma-network.com	✓	✓	✓	✓
		peer1.manufacturer.pharma-network.com	✗	✓	✓	✗
Distributor	ca.distributor.pharma-network.com	peer0.distributor.pharma-network.com	✓	✓	✓	✓
		peer1.distributor.pharma-network.com	✗	✓	✓	✗
Retailer	ca.retailer.pharma-network.com	peer0.retailer.pharma-network.com	✓	✓	✓	✓
		peer1.retailer.pharma-network.com	✗	✓	✓	✗
Consumer	ca.consumer.pharma-network.com	peer0.consumer.pharma-network.com	✓	✓	✓	✓
		peer1.consumer.pharma-network.com	✗	✓	✓	✗
Transporter	ca.transporter.pharma-network.com	peer0.transporter.pharma-network.com	✓	✓	✓	✓
		peer1.transporter.pharma-network.com	✗	✓	✓	✗

3. PROJECT STRUCTURE

- The project is built from scratch using “**Atom**” as the editor.
- Entire **pharma-net** project is organized into below listed main folders -
 - network
 - chaincode
 - application
 - test
- There is an additional “chaincode-advanced” folder which is empty and only used during docker volume mappings

The project folders are explained in further sections.

3.1. NETWORK

network folder contains all the artifacts related to Docker network setup. The structure and files available in this folder are outlined below.

Folder/ File Name	Purpose
Bin	Contains all the fabric binaries required during network setup phase
channel-artifacts	Channel artifacts such as orderer genesis file, channel creation and anchor peer updates are stored in this folder during network setup
docker-base	Contains base Docker compose YAML files for orderer and peer configurations
Scripts	Has network automation script modules used during network setup
configtx.yaml	Configuration file to generate channel artifacts
crypto-config.yaml	Configuration file to generate crypto materials
docker-compose.yml	Main Docker compose YAML file for the pharma network
fabricNetwork.sh	Network automation shell script

3.2. CHAINCODE

chaincode folder contains the smart contracts with core business logic of pharma network.
pharmanet chaincode is comprised of below listed three smart contracts-

- **registrationcontract:**
 - Instantiating chaincode on the network
 - Register new company
 - Add new drug
- **transferdrugcontract:**
 - Create purchase order
 - Create shipment
 - Update shipment
 - Retail drug
- **viewlifecyclecontract:**
 - View drug history
 - View drug current state Structure of chaincode folder is explained below.

Folder/ File Name	Purpose
Contracts	Contains the three contracts that are part of pharmanet chaincode
Enums	Enum objects used across all smart contracts
Utils	Helper files with common functions for ledger operations
index.js	Main entry file for the NodeJS smart contract
package.json	JSON configuration file listing all the dependencies and scripts
package-lock.json	Auto generated file from "npm install" command

3.3. APPLICATION

Contains code for the Node.JS application to

- Submit transactions on the network by invoking respective smart contracts functions
- Create node server and expose API end points for accessing from Postman

Structure of this folder is explained below

Folder/ File Name	Purpose	Sub Folder	Sub Folder Purpose
ccps	Common Connection Profiles for each organization on the network		
enums	Enum objects used throughout the application		
helpers	Helper files for establishing gateway connections to network and file system functions		
identity	Folder where auto-generated wallets for each organization are stored		
modules	Node.JS modules for submitting transactions on fabric network	common	Common modules to auto generate org wallets
		registration	Modules to submit transactions to registration smart contract
		transferdrug	Modules to submit transactions to transfer drug smart contract
		viewlifecycle	Modules to submit transactions to view lifecycle smart contract
routes	HTTP request handlers for each of the Node.JS modules		
index.js	Main entry point for the Node.js application		
package.json	JSON configuration file listing all the dependencies		

3.4. TEST

Contains postman collection files for initiating test HTTP requests through postman -

- History.postman_collection.json:** Collection file for History Track Down test cases
- Initiation.postman_collection.json:** Collection file for Initiation test cases
- Supply Chain.postman_collection.json:** Collection file for Supply Chaintest cases

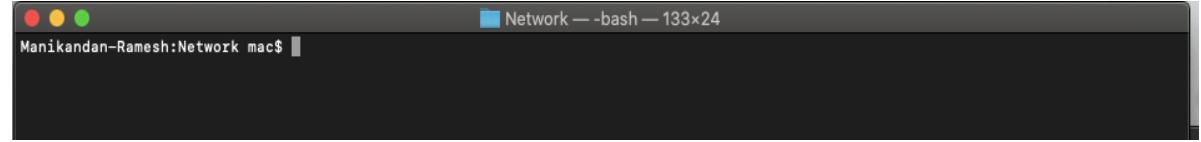
4. PRE SETUP STEPS

This section details the steps to be performed before starting network setup

- Step 1:** Download zip folder from Google Drive link available in assignment submission page.
- Step 2:** Unzip the folder to a suitable location on Ubuntu machine.
- Step 3:** Navigate to “**pharma-net > network**” path inside unzipped folder
- Step 4:** Replace bin folder files based on Operating System used, this .zip folder contains bin files for OS X (Mac -OS), if need to run in Linux need to change bin folder files

Note: - Given steps below are mentioned as per in Mac-OS, and its practice.

This will open Terminal pointing to network folder as shown below

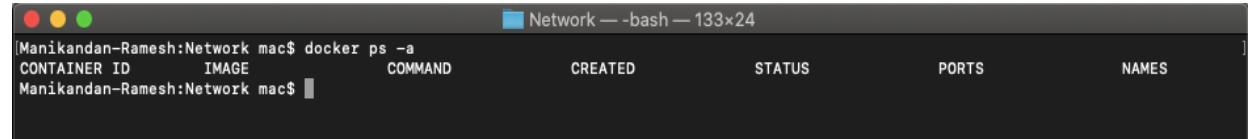


```
Network -- bash -- 133x24
Manikandan-Ramesh:Network mac$ docker ps -a
```

- Step 5:** Execute below command and ensure that no Docker containers are running.

```
docker ps -a
```

The output of above command should look similar to shown in below screenshot:



```
Network -- bash -- 133x24
Manikandan-Ramesh:Network mac$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
Manikandan-Ramesh:Network mac$
```

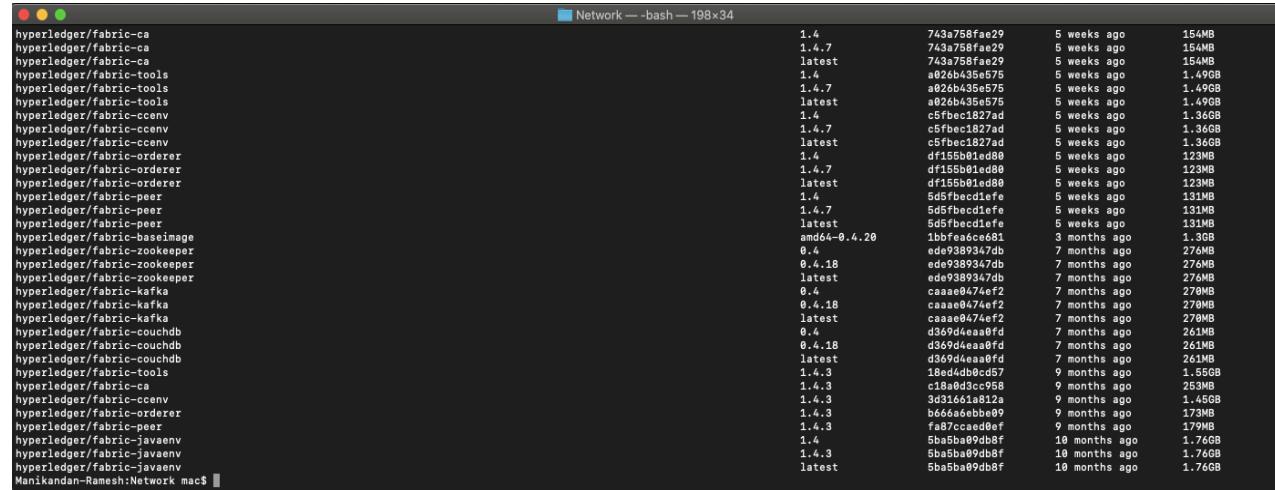
In case any containers are running, use below command to force kill them and re-run “`docker ps -a`” command to ensure all containers have been removed

```
docker rm -f $(docker ps -aq)
```

- Step 6:** Run below command in the terminal to ensure no stale docker images are available

```
docker images
```

The output of this command should look like below:



```
Network -- bash -- 198x34
hyperledger/fabric-ca          1.4                  743a758fae29   5 weeks ago    154MB
hyperledger/fabric-ca          1.4.7                743a758fae29   5 weeks ago    154MB
hyperledger/fabric-ca          latest               743a758fae29   5 weeks ago    154MB
hyperledger/fabric-tools       1.4                  a026b435e675   5 weeks ago    1.49GB
hyperledger/fabric-tools       1.4.7                a026b435e675   5 weeks ago    1.49GB
hyperledger/fabric-tools       latest               a026b435e675   5 weeks ago    1.49GB
hyperledger/fabric-ccenv       1.4                  c5fbec1827ad   5 weeks ago    1.36GB
hyperledger/fabric-ccenv       1.4.7                c5fbec1827ad   5 weeks ago    1.36GB
hyperledger/fabric-ccenv       latest               c5fbec1827ad   5 weeks ago    1.36GB
hyperledger/fabric-peer        1.4                  df155b01e098   5 weeks ago    12MB
hyperledger/fabric-peer        1.4.7                df155b01e098   5 weeks ago    12MB
hyperledger/fabric-peer        latest               df155b01e098   5 weeks ago    12MB
hyperledger/fabric-orderer     1.4                  df155b01e098   5 weeks ago    123MB
hyperledger/fabric-orderer     1.4.7                df155b01e098   5 weeks ago    131MB
hyperledger/fabric-peer        1.4                  5d5fbecd1fe   5 weeks ago    131MB
hyperledger/fabric-peer        1.4.7                5d5fbecd1fe   5 weeks ago    131MB
hyperledger/fabric-peer        latest               5d5fbecd1fe   5 weeks ago    131MB
hyperledger/fabric-baseimage   amd64-0.4.20      1bbfea6c6e681   3 months ago   1.3GB
hyperledger/fabric-zookeeper  0.4                  ede9389347db   7 months ago   276MB
hyperledger/fabric-zookeeper  0.4.18               ede9389347db   7 months ago   276MB
hyperledger/fabric-zookeeper  latest               ede9389347db   7 months ago   276MB
hyperledger/fabric-kafka       0.4                  caaae0474ef2   7 months ago   270MB
hyperledger/fabric-kafka       0.4.18               caaae0474ef2   7 months ago   270MB
hyperledger/fabric-kafka       latest               caaae0474ef2   7 months ago   270MB
hyperledger/fabric-couchdb    0.4                  d369d4ea0afdf  7 months ago   261MB
hyperledger/fabric-couchdb    0.4.18               d369d4ea0afdf  7 months ago   261MB
hyperledger/fabric-tools       1.4.3                18ed4db0cd57   9 months ago   1.55GB
hyperledger/fabric-ca          1.4.3                c18a0d3cc958   9 months ago   253MB
hyperledger/fabric-ccenv       1.4.3                3d31661a812a   9 months ago   1.45GB
hyperledger/fabric-orderer     1.4.3                b66a6ebbe09   9 months ago   173MB
hyperledger/fabric-peer        1.4.3                fa87ccaaed0ef 9 months ago   179MB
hyperledger/fabric-javavenv   1.4                  5ba5ba09db8f   10 months ago  1.76GB
hyperledger/fabric-javavenv   1.4.3               5ba5ba09db8f   10 months ago  1.76GB
hyperledger/fabric-javavenv   latest               5ba5ba09db8f   10 months ago  1.76GB
Manikandan-Ramesh:Network mac$
```

In case the command shows any image with repository like “**dev-peer*.*.pharma-network.com-***”, use below command to remove these images (Substitute **ImageID** with “Image ID” available from above command for dev images) - see below screenshots for reference. This step is required to ensure no stale code is left over from any previous runs before starting network setup.

```
docker rmi ImageID
```

```

Network -- bash — 198x34
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
dev-peer0.manufacturer.pharma-network.com-pharmanet-1.1-941d9f41fb3927bb09dd9345376e89e8c838651f9ca9196ab5ff6d70be099b48  latest   e77ef34e942b  43 hours ago  1.37GB
hyperledger/fabric-ca                                         1.4      743a758fae29  5 weeks ago   154MB
hyperledger/fabric-ca                                         1.4.7    743a758fae29  5 weeks ago   154MB
hyperledger/fabric-tools                                     latest   743a758fae29  5 weeks ago   154MB
hyperledger/fabric-tools                                     1.4     a026b435e575  5 weeks ago   1.49GB
hyperledger/fabric-tools                                     1.4.7    a026b435e575  5 weeks ago   1.49GB

Network -- bash — 198x7
Manikandan-Ramesh:Network mac$ docker rmi e77ef34e942b
Untagged: dev-peer0.manufacturer.pharma-network.com-pharmanet-1.1-941d9f41fb3927bb09dd9345376e89e8c838651f9ca9196ab5ff6d70be099b48:latest
Deleted: sha256:e77ef34e942b1b8474261fad78949ddbc3975a82d2571ba4357ea2af9e074d4b
Deleted: sha256:b59b187f1b5e26ec9b25b9e06bc77921dcce3ba30e628afad92693a7dac97c
Deleted: sha256:3b5b1aa2f67a76929340741839167259e976494684399db8effe30c80a4cbc
Deleted: sha256:ed60eca263e2232f5af484484413b2382043a8e0b8c16e95b8e2be77b8a65581
Manikandan-Ramesh:Network mac$ 

```

- **Step 7:** Run below commands one by one in terminal to give necessary permissions on fabric binaries required during network setup

```

Network -- bash — 198x7
[Manikandan-Ramesh:Network mac$ export PATH=~/desktop/Capstone_Project_Manikandan/pharma-net/network/bin:$PATH
[Manikandan-Ramesh:Network mac$ chmod +x fabricNetwork.sh
Manikandan-Ramesh:Network mac$ 

```

- **Step 8:** Run below commands in terminal one by one to give necessary permissions on all files inside project folder

```

[Manikandan-Ramesh:Network mac$ cd scripts/
[Manikandan-Ramesh:scripts mac$ chmod +x installChaincode.sh
[Manikandan-Ramesh:scripts mac$ chmod +x updateChaincode.sh
[Manikandan-Ramesh:scripts mac$ chmod +x utils.sh
[Manikandan-Ramesh:scripts mac$ chmod +x bootstrap.sh
Manikandan-Ramesh:scripts mac$ 

```

- With this all pre-setup steps are completed and network setup can be started, need to move back to network folder

```

[Manikandan-Ramesh:scripts mac$ cd -
/Users/manikandan/desktop/Capstone_Project_Manikandan/pharma-net/Network
Manikandan-Ramesh:Network mac$ 

```

5. BOOTSTRAP PHARMA-NETWORK

- Starting pharma-network is a one step process by executing a single command which triggers the network automation script. The command performs following steps to bootstrap pharma-network:
 - Generates crypto-materials for all the 5 organizations
 - Generates channel artifacts for -
 - Orderer genesis block
 - Channel creation
 - Anchor peer updates for each org
 - Starts Docker containers for -
 - Orderer
 - 10 peers, 2 per org
 - 5 CAs, 1 per org
 - cli
 - Creates channel “**pharmachannel**”
 - Joins all 10 peers of 5 orgs to pharmachannel
 - Updates peer0 as Anchor peer for each org
- Run the following command to bootstrap pharma network inside terminal window

```
./fabricNetwork.sh generate
```

- Command will ask confirmation as shown below. Simply press Enter to allow command to proceed.

```
[Manikandan-Ramesh:Network mac$ ./fabricNetwork.sh generate
Generating certs and genesis block for channel 'pharmachannel' with CLI timeout of '15' seconds and CLI delay of '5' seconds and chaincode version '1.1'
Continue? [Y/n] Y
```

- Wait for command to complete certificate generation . Screenshots of terminal at different stages of command execution are shown below.

```
#####
##### Generate certificates using cryptogen tool #####
#####
+ cryptogen generate --config=./crypto-config.yaml
manufacturer.pharma-network.com
distributor.pharma-network.com
retailer.pharma-network.com
consumer.pharma-network.com
transporter.pharma-network.com
+ res=0
+ set +x

/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Network/bin/configtxgen
#####
##### Generating Orderer Genesis block #####
#####
+ configtxgen -profile OrdererGenesis -channelID pharma-sys-channel -outputBlock ./channel-artifacts/genesis.block
2020-06-19 20:49:57.773 IST [common.tools.configtxgen] main > INFO 001 Loading configuration
2020-06-19 20:49:57.799 IST [common.tools.configtxgen.localconfig] completeInitialization > INFO 002 orderer type: solo
2020-06-19 20:49:57.799 IST [common.tools.configtxgen.localconfig] Load > INFO 003 Loaded configuration: /Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Network/configtx.yaml
2020-06-19 20:49:57.818 IST [common.tools.configtxgen.localconfig] completeInitialization > INFO 004 orderer type: solo
2020-06-19 20:49:57.818 IST [common.tools.configtxgen.localconfig] LoadTopLevel > INFO 005 Loaded configuration: /Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Network/configtx.yaml
2020-06-19 20:49:57.837 IST [common.tools.configtxgen] doOutputBlock > INFO 006 Generating genesis block
2020-06-19 20:49:57.837 IST [common.tools.configtxgen] doOutputBlock > INFO 007 Writing genesis block
+ res=0
```

- Run the following command to bootstrap pharma network inside terminal window

```
./fabricNetwork.sh up
```

- Command will ask confirmation as shown below. Simply press Enter to allow command to proceed.

```
Manikandan-Ramesh:Network mac$ ./fabricNetwork.sh up
Starting for channel 'pharmachannel' with CLI timeout of '15' seconds and CLI delay of '5' seconds and chaincode version '1.1'
Continue? [Y/n] Y
```

- Wait for command to complete Network Setup . Screenshots of terminal at different stages of command execution are shown below.

```
Creating volume "network_peer1.manufacturer.pharma-network.com" with default driver
Creating volume "network_peer0.distributor.pharma-network.com" with default driver
Creating volume "network_peer1.distributor.pharma-network.com" with default driver
Creating volume "network_peer0.retailer.pharma-network.com" with default driver
Creating volume "network_peer1.retailer.pharma-network.com" with default driver
Creating volume "network_peer0.consumer.pharma-network.com" with default driver
Creating volume "network_peer1.consumer.pharma-network.com" with default driver
Creating volume "network_peer0.transporter.pharma-network.com" with default driver
Creating volume "network_peer1.transporter.pharma-network.com" with default driver
Creating orderer.pharma-network.com    ... done
Creating peer1.distributor.pharma-network.com ... done
Creating peer0.transporter.pharma-network.com ... done
Creating ca.consumer.pharma-network.com ... done
Creating chaincode
Creating ca.manufacturer.pharma-network.com ... done
Creating peer1.manufacturer.pharma-network.com ... done
Creating peer0.distributor.pharma-network.com ... done
Creating peer0.consumer.pharma-network.com ... done
Creating peer1.transporter.pharma-network.com ... done
Creating peer0.manufacturer.pharma-network.com ... done
Creating peer0.retailer.pharma-network.com ... done
Creating ca.distributor.pharma-network.com ... done
Creating peer1.consumer.pharma-network.com ... done
Creating ca.retailer.pharma-network.com ... done
Creating peer1.retailer.pharma-network.com ... done
Creating ca.transporter.pharma-network.com ... done
Creating cli ... done
```

```
Setting Up Hyperledger Fabric Network

Channel name : pharmachannel
Creating channel...
+ peer channel create -o orderer.pharma-network.com:7050 -c pharmachannel -f ./channel-artifacts/channel.tx
+ res=0
+ set +x
2020-06-19 15:20:28.993 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-06-19 15:20:29.063 UTC [cli.common] readBlock -> INFO 002 Received block: 0
===== Channel 'pharmachannel' created =====

Having all peers join the channel...
+ peer channel join -b pharmachannel.block
+ res=0
+ set +x
2020-06-19 15:20:29.319 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-06-19 15:20:29.409 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
===== peer0.manufacturer.pharma-network.com joined channel 'pharmachannel' =====

+ peer channel join -b pharmachannel.block
+ res=0
```

```
+ peer channel join -b pharmachannel.block
+ res=0
+ set +x
2020-06-19 15:20:45.101 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-06-19 15:20:45.184 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
===== peer1.distributor.pharma-network.com joined channel 'pharmachannel' =====

+ peer channel join -b pharmachannel.block
+ res=0
+ set +x
2020-06-19 15:20:50.350 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-06-19 15:20:50.445 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
===== peer0.retailer.pharma-network.com joined channel 'pharmachannel' =====

+ peer channel join -b pharmachannel.block
+ res=0
+ set +x
2020-06-19 15:20:55.661 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-06-19 15:20:55.729 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
===== peer1.retailer.pharma-network.com joined channel 'pharmachannel' =====

+ peer channel join -b pharmachannel.block
+ res=0
+ set +x
2020-06-19 15:21:00.928 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-06-19 15:21:01.002 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

Successful completion will look like shown below.

```

Updating anchor peers for consumer...
+ peer channel update -o orderer.pharma-network.com:7050 -c pharmachannel -f ./channel-artifacts/consumerMSPanchors.tx
+ res=0
+ set +x
2020-06-19 15:21:37.701 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-06-19 15:21:37.736 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
===== Anchor peers updated for org 'consumerMSP' on channel 'pharmachannel' =====

Updating anchor peers for transporter...
+ peer channel update -o orderer.pharma-network.com:7050 -c pharmachannel -f ./channel-artifacts/transporterMSPanchors.tx
+ res=0
+ set +x
2020-06-19 15:21:42.927 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-06-19 15:21:42.959 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
===== Anchor peers updated for org 'transporterMSP' on channel 'pharmachannel' =====

===== All GOOD, Hyperledger Fabric Pharma Network Is Now Up and Running! =====

[----] [ \ ] [ - ]
[ - ] [ M ] [ - ]
[ - ] [ \ ] [ - ]
[----] [ | ] [ - ]

Manikandan-Ramesh:Network mac$ █

```

- Run below command and ensure that status of all docker containers is showing “Up xx seconds” or “Up xx minutes” to confirm that network setup was successful

```
docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
eb48d5d8b9fa2	hyperledger/fabric-tools:latest	"/bin/bash"	19 minutes ago	Up 19 minutes	0.0.0.0:9054->9054/tcp	cli
6f86e5538d9d	hyperledger/fabric-ca:latest	"sh -c 'fabric-ca-se_'"	19 minutes ago	Up 19 minutes	0.0.0.0:7054->7054/tcp	ca.retailer.pharma-network.c
9ef2c5f09d6	hyperledger/fabric-ca:latest	"sh -c 'fabric-ca-se_'"	19 minutes ago	Up 19 minutes	0.0.0.0:7051->7051/tcp, 0.0.0.0:12053->12053/tcp	ca.manufacturer.pharma-netwo
rk.com	de2027595369	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	0.0.0.0:12051->12051/tcp, 0.0.0.0:12053->12053/tcp	peer1.retailer.pharma-netw
030a35f9b627	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	Up 19 minutes	0.0.0.0:7051->7051/tcp, 0.0.0.0:7053->7053/tcp	peer0.manufacturer.pharma-ne
twork.com	5c1ad7c82fd7	hyperledger/fabric-ca:latest	"sh -c 'fabric-ca-se_'"	19 minutes ago	0.0.0.0:8054->8054/tcp	ca.distributor.pharma-networ
k.com	3d6e4ab974ff	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	0.0.0.0:11051->11051/tcp, 0.0.0.0:11053->11053/tcp	peer0.retailer.pharma-networ
kb.com	eb6f7e909e28	hyperledger/fabric-ca:latest	"sh -c 'fabric-ca-se_'"	19 minutes ago	0.0.0.0:10054->7054/tcp	ca.consumer.pharma-network.c
on	17bc0088e1d7	hyperledger/fabric-peer:latest	"sh -c 'fabric-ca-se_'"	19 minutes ago	0.0.0.0:11054->7054/tcp	ca.transporter.pharma-networ
k.com	3d1da53b97a7	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	0.0.0.0:14051->14051/tcp, 0.0.0.0:14053->14053/tcp	peer1.consumer.pharma-networ
dad15bcead72	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	Up 19 minutes	0.0.0.0:8051->8051/tcp, 0.0.0.0:8053->8053/tcp	peer1.manufacturer.pharma-ne
twork.com	90f198619d26	hyperledger/fabric-ccenv:latest	"/bin/bash -c 'sleep_'"	19 minutes ago	Up 19 minutes	chaincode
c0feb69eae1	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	Up 19 minutes	0.0.0.0:16051->16051/tcp, 0.0.0.0:16053->16053/tcp	peer1.transporter.pharma-net
work.com	e76d5abf4cfb	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	0.0.0.0:13051->13051/tcp, 0.0.0.0:13053->13053/tcp	peer0.consumer.pharma-netwo
k.com	96df1fcf34624	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	0.0.0.0:15051->15051/tcp, 0.0.0.0:15053->15053/tcp	peer0.transporter.pharma-net
work.com	000005b540f5	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	0.0.0.0:10051->10051/tcp, 0.0.0.0:10053->10053/tcp	peer1.distributor.pharma-net
work.com	39d2fdc38a9	hyperledger/fabric-peer:latest	"peer node start"	19 minutes ago	0.0.0.0:9051->9051/tcp, 0.0.0.0:9053->9053/tcp	peer0.distributor.pharma-net
work.com	423f8c8c981c	hyperledger/fabric-orderer:latest	"orderer"	19 minutes ago	0.0.0.0:7050->7050/tcp	orderer.pharma-network.com

- In case of any time-out error while running network setup command, first ensure that all docker containers are up using above command and then rerun “./fabricNetwork.sh up” to complete network setup.

Note that pharma-network is started in Production mode though chaincode container is created as part of network setup.

6. INSTALL AND INSTANTIATE CHAINCODE

6.1. INSTALL AND INSTANTIATE PHARMANET CHAINCODE

- Chaincode “**pharmenet**” can be installed and instantiated by running a single command using network automation script.
- This command performs following actions:
 - Installs chaincode one both peers of every organization
 - Instantiates chaincode on “pharmachannel” through peer0 of “Manufacturer” org
- Run following command on same terminal window opened above step to install and instantiate chaincode

```
./fabricNetwork.sh install
```

- Command will ask confirmation as shown below. Simply press Enter to allow command to proceed.

```
Installing chaincode for channel 'pharmachannel' with CLI timeout of '15' seconds and CLI delay of '5' seconds
and chaincode version '1.1'
Continue? [Y/n] ■
```

- Wait for command to complete chaincode installation and instantiation. Screenshots of terminal at different stages of command execution are shown below.

```
Manikandan-Ramesh:Network mac$ ./fabricNetwork.sh install
Installing chaincode for channel 'pharmachannel' with CLI timeout of '15' seconds and CLI delay of '5' seconds
and chaincode version '1.1'
Continue? [Y/n] Y
proceeding ...
LOCAL_VERSION=1.4.3
DOCKER_IMAGE_VERSION=1.4.7
===== WARNING =====
  Local fabric binaries and docker images are
  out of sync. This may cause problems.
=====

<---| |---| / \ |---\| |---|
 \--\| | | / \ \ |---\| | |
 |---/| | | / \ \ |---\| | |
 
Deploying Chaincode PHARMANET On Pharma Network

Channel name : pharmachannel
Installing chaincode on all peers...
Installing chaincode on peer0.manufacturer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:42:56.365 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:42:56.366 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:42:56.474 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:
"OK" >
===== Chaincode is installed on peer0.manufacturer =====

Installing chaincode on peer1.manufacturer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:01.758 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:01.758 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:01.880 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:
"OK" >
```

```

Manikandan-Ramesh:Network mac$ ./fabricNetwork.sh install
Installing chaincode for channel 'pharmachannel' with CLI timeout of '15' seconds and CLI delay of '5' seconds
and chaincode version '1.1'
Container? [Y/n] .
proceeding ...
LOCAL_VERSION=1..3
DOCKER_IMAGE_VERSION=1.4.7
=====
        WARNING
        Local fabric binaries and docker images are
        out of sync. This may cause problems.
=====

Deploying Chaincode PHARMANET On Pharma Network
Channel name : pharmachannel
Installing chaincode on all peers...
Installing chaincode on peer0.manufacturer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:42:56.365 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:42:56.366 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:42:56.474 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:
"OK" >
===== Chaincode is installed on peer0.manufacturer =====
Installing chaincode on peer1.manufacturer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:01.758 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:01.758 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:01.880 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:
"OK" >
===== Chaincode is installed on peer1.manufacturer =====
Installing chaincode on peer0.distributor.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:07.126 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:07.126 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:07.222 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer0.distributor =====
Installing chaincode on peer1.distributor.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:12.425 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:12.425 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:12.568 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer1.distributor =====
Installing chaincode on peer0.retailer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:17.867 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:17.868 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:17.976 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer0.retailer =====
Installing chaincode on peer1.retailer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:23.226 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:23.226 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:23.366 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer1.retailer =====
+ set +x
2020-06-19 15:43:23.226 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:23.226 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:23.366 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer1.retailer =====
Installing chaincode on peer0.consumer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:28.681 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:28.681 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:28.787 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer0.consumer =====
Installing chaincode on peer1.consumer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:34.118 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:34.118 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:34.251 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer1.consumer =====
Installing chaincode on peer0.transporter.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:39.595 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:39.595 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:39.774 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer0.transporter =====
Installing chaincode on peer1.transporter.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chai
ncode/
+ res=0
+ set +x
2020-06-19 15:43:44.996 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:44.996 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:45.085 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer1.transporter =====

```

- Successful completion will look like shown below.

```

Installing chaincode on peer1.transporter.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.1 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chaincode/
+ res=0
+ set +x
2020-06-19 15:43:44.996 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:44.996 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-06-19 15:43:45.085 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
=====
Chaincode is installed on peer1.transporter =====

Instantiating chaincode on channel using peer0.manufacturer.pharma-network.com...
+ peer chaincode instantiate -o orderer.pharma-network.com:7050 -C pharmachannel -n pharmanet -l node -v 1.1 -c '{"Args":["org.pharma
-network.pharmanet.registrationcontract:instantiate"]}' -P 'OR ('\\'manufacturerMSP.member'\\', '\\'distributorMSP.member'\\', '\\'retai
llerMSP.member'\\', '\\'consumerMSP.member'\\', '\\'transporterMSP.member'\\')'
+ res=0
+ set +x
2020-06-19 15:43:50.434 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-06-19 15:43:50.434 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
=====
Chaincode is instantiated on peer0.manufacturer on channel 'pharmachannel' =====

=====
All GOOD, Chaincode PHARMANET Is Now Installed & Instantiated On Pharma Network =====

[END]
Manikandan-Ramesh:Network mac$ 

```

Note that depending on machine configuration and internet speed, chaincode instantiation can take between 5 to 10 minutes to complete since peer0 of manufacturer org builds a new chaincode container where all dependencies of code are first installed through “npm install” before chaincode gets instantiated.

6.2. VERIFY CHAINCODE INSTANTIATION

- To **re-confirm** that chaincode was properly instantiated, open a new tab in terminal to view logs of peer. In this new tab, type infollowing command:

```
docker logs -f peer0.manufacturer.pharma-network.com
```

- After pressing Enter, the output should be similar to shown below to confirm that chaincode instantiation was successful

```

tion=82.595036ms
2020-06-19 15:21:44.422 UTC [gossip.comm] sendToEndpoint -> WARN 132 Failed obtaining connection for peer1.consumer.pharma-network.com:140051, PKIid:d3efbc17ab65e8d782462762697b850712b0e0b6f730aaf4499e774c0e6d8467 reason: context deadline exceeded
2020-06-19 15:21:44.422 UTC [gossip.discovery] expireDeadMembers -> WARN 133 Entering [d3efbc17ab65e8d782462762697b850712b0e0b6f730aa
f4499e774c0e6d8467]
2020-06-19 15:21:44.422 UTC [gossip.discovery] expireDeadMembers -> WARN 134 Closing connection to Endpoint: peer1.consumer.pharma-ne
twork.com:140051, InternalEndpoint: , PKI-ID: d3efbc17ab65e8d782462762697b850712b0e0b6f730aaf4499e774c0e6d8467, Metadata:
2020-06-19 15:21:44.631 UTC [comm.grpc.server] 1 -> INFO 136 unary call completed grpc.service=gossip.Gossip grpc.method=Ping grpc.re
quest_deadline=2020-06-19T15:21:46.631Z grpc.peer_address=172.19.0.5:45542 grpc.code=OK grpc.call_duration=60.673µs
2020-06-19 15:21:46.260 UTC [comm.grpc.server] 1 -> INFO 137 unary call completed grpc.service=gossip.Gossip grpc.method=Ping grpc.re
quest_deadline=2020-06-19T15:21:48.26Z grpc.peer_address=172.19.0.14:50726 grpc.code=OK grpc.call_duration=91.783µs
2020-06-19 15:21:46.388 UTC [comm.grpc.server] 1 -> INFO 138 unary call completed grpc.service=gossip.Gossip grpc.method=Ping grpc.re
quest_deadline=2020-06-19T15:21:48.388Z grpc.peer_address=172.19.0.13:41454 grpc.code=OK grpc.call_duration=134.923µs
2020-06-19 15:42:56.468 UTC [endorser] callChaincode -> INFO 139 []@[0f593713] Entry chaincode: name:"lscc"
2020-06-19 15:42:56.470 UTC [lscc] executeInstall -> INFO 13a Installed Chaincode [pharmanet] Version [1.1] to peer
2020-06-19 15:42:56.471 UTC [endorser] callChaincode -> INFO 13b []@[0f593713] Exit chaincode: name:"lscc" (2ms)
2020-06-19 15:42:56.472 UTC [comm.grpc.server] 1 -> INFO 13c unary call completed grpc.service=protos.Endorser grpc.method=ProcessPro
posal grpc.peer_address=172.19.0.19:56642 grpc.code=OK grpc.call_duration=4.25516ms
2020-06-19 15:43:50.448 UTC [endorser] callChaincode -> INFO 13d [pharmacchannel][0904cc5c] Entry chaincode: name:"lscc"
2020-06-19 15:45:31.018 UTC [endorser] callChaincode -> INFO 13e [pharmacchannel][0904cc5c] Exit chaincode: name:"lscc" (100626ms)
2020-06-19 15:45:31.029 UTC [comm.grpc.server] 1 -> INFO 13f unary call completed grpc.service=protos.Endorser grpc.method=ProcessPro
posal grpc.peer_address=172.19.0.19:56682 grpc.code=OK grpc.call_duration=1m40.63876678s
2020-06-19 15:45:33.170 UTC [gossip.privdata] StoreBlock -> INFO 140 [pharmachannel] Received block [6] from buffer
2020-06-19 15:45:33.207 UTC [committer.txvalidator] Validate -> INFO 141 [pharmacchannel] Validated block [6] in 37ms
2020-06-19 15:45:33.215 UTC [cceventmgmt] HandleStateUpdates -> INFO 142 Channel [pharmacchannel]: Handling deploy or update of chainc
ode [pharmanet]
2020-06-19 15:45:33.302 UTC [kvledger] CommitWithPvtData -> INFO 143 [pharmachannel] Committed block [6] with 1 transaction(s) in 90m
s (state_validation=30ms block_and_pvtdata_commit=49ms state_commit=4ms) commitHash=[ff76940da8e424b30b6f42cafceef7ead2dd8d9209ad6e49
fb515972451ad4cd]

```

- Press Ctrl + C on terminal after verifying peer logs to exit from logs and then close the terminal tab.

7. START NODE APPLICATION

7.1. INSTALL DEPENDENCIES

Before starting node application, dependencies need to be installed by following below steps

- Open a new terminal tab and move to project location till application folder and run following command to verify if npm is installed on VM

```
npm -v
```

```
[Manikandan-Ramesh:~ mac$ npm -v  
6.12.1  
Manikandan-Ramesh:~ mac$ ]
```

If npm is installed, this command will display the version like shown below

- If above command does not show npm version, run below commands one by one on terminal to install npm

```
npm install
```

Screen will look like below after above npm install completes (it can take few minutes to complete execution)

```
npm WARN notsup Unsupported engine for fabric-network@1.4.9: wanted: {"node":">=10.15.3 || ^12.15.0", "npm":">=6.4.1"} (current: {"node":"12.13.1", "npm":"6.12.1"})  
npm WARN notsup Not compatible with your version of node/npm: fabric-network@1.4.9  
npm WARN notsup Unsupported engine for fabric-ca-client@1.4.9: wanted: {"node":">=10.15.3 || ^12.15.0", "npm":">=6.4.1"} (current: {"node":"12.13.1", "npm":"6.12.1"})  
npm WARN notsup Not compatible with your version of node/npm: fabric-ca-client@1.4.9  
npm WARN notsup Unsupported engine for fabric-client@1.4.9: wanted: {"node":">=10.15.3 || ^12.15.0", "npm":">=6.4.1"} (current: {"node":"12.13.1", "npm":"6.12.1"})  
npm WARN notsup Not compatible with your version of node/npm: fabric-client@1.4.9  
npm WARN acorn-jsx@5.2.0 requires a peer of acorn@^6.0.0 || ^7.0.0 but none is installed. You must install peer dependencies yourself.  
npm WARN pharma_network@1.0.0 No repository field.  
npm WARN pharma_network@1.0.0 No license field.  
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: pkcs11js@1.0.21 (node_modules/pkcs11js):  
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: pkcs11js@1.0.21 install: `node-gyp rebuild`  
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: Exit status 1  
  
added 421 packages from 384 contributors and audited 423 packages in 39.074s  
found 15 low severity vulnerabilities  
  run `npm audit fix` to fix them, or `npm audit` for details  
  
New minor version of npm available! 6.12.1 => 6.14.5  
Changelog: https://github.com/npm/cli/releases/tag/v6.14.5  
Run npm install -g npm to update!
```

- To confirm npm install was successful, open “application” folder on file explorer and observe that a new “node_modules” folder is created

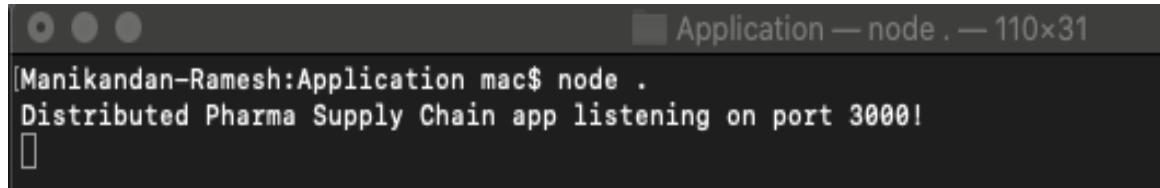


7.2. START NODE APPLICATION

- Run below command inside application folder on terminal to start the node application

```
node .
```

- Successful execution of the command will look like shown below



A screenshot of a terminal window titled "Application — node . — 110x31". The window shows the command "node ." being run and its output: "Manikandan-Ramesh:Application mac\$ node . Distributed Pharma Supply Chain app listening on port 3000!". The terminal has three tabs open at the top.

```
[Manikandan-Ramesh:Application mac$ node .
Distributed Pharma Supply Chain app listening on port 3000!]
```

- This indicates that the application is now ready to accept HTTP requests on port 3000

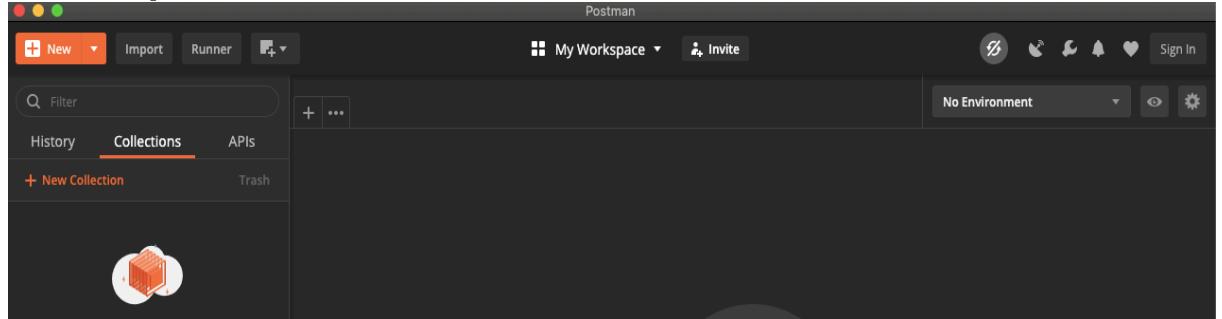
Note that “identity” folder which stores various organization wallets is initially empty and wallets get created automatically as requests are submitted for each organization

8. TESTING THROUGH POSTMAN COLLECTION FILES

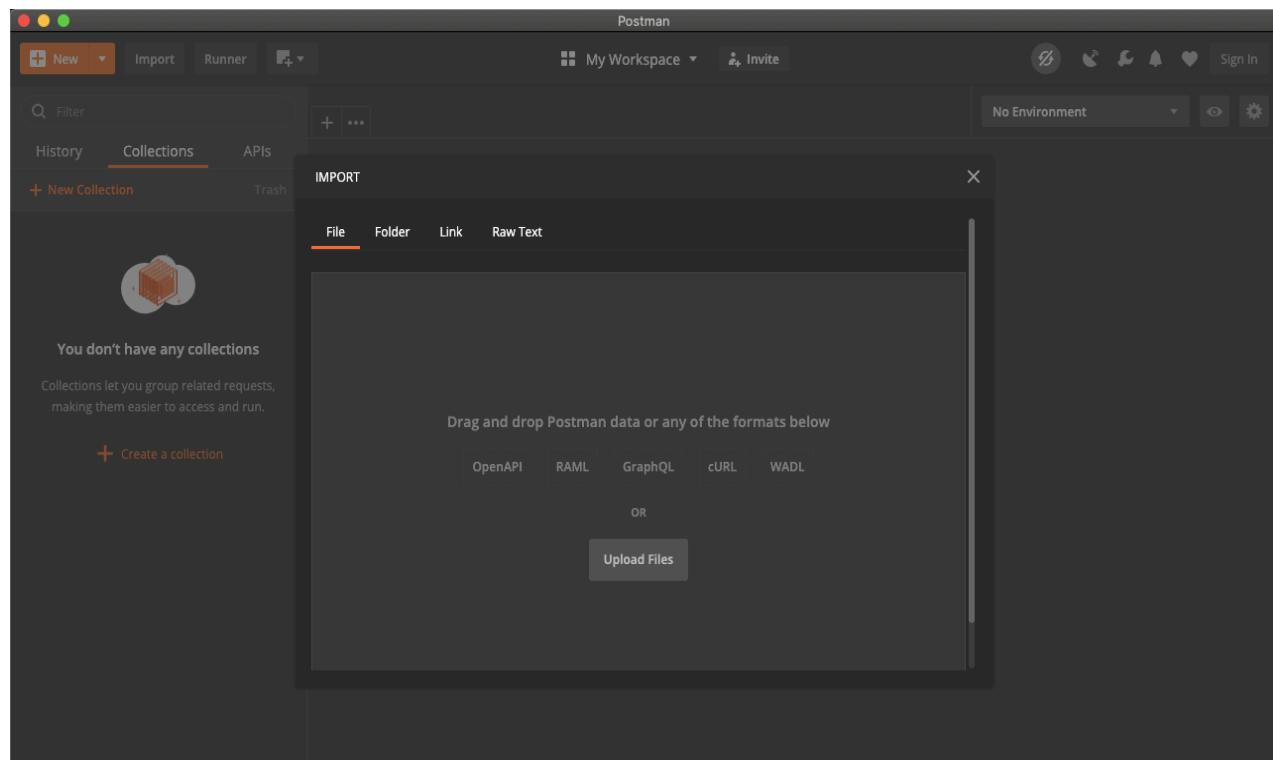
This section explains the steps to be followed to validate various smart contract transactions available in pharma-network by making HTTP API calls through Postman

8.1. IMPORT POSTMAN COLLECTION FILES

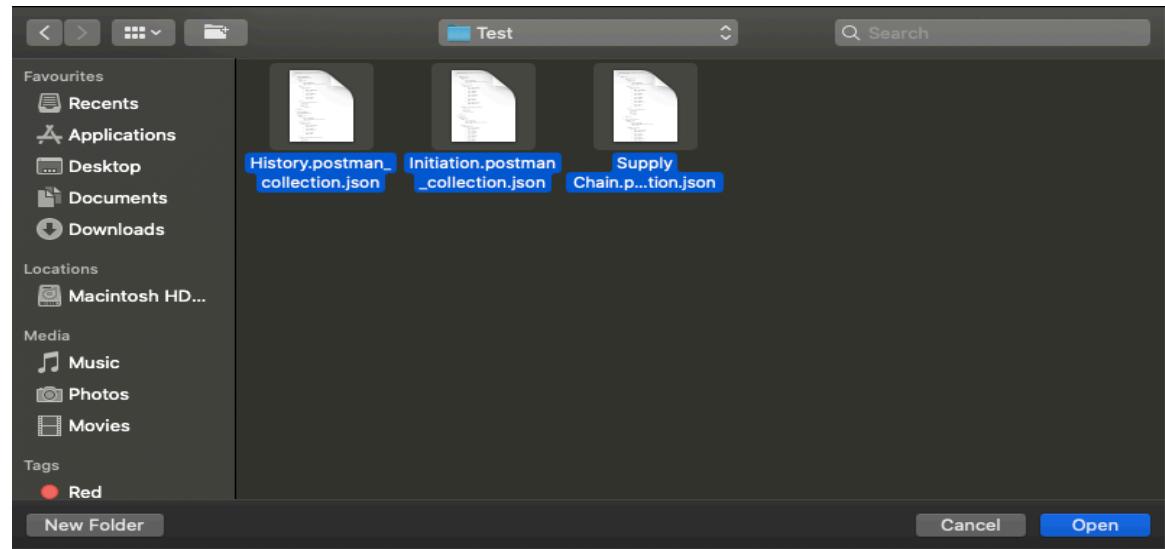
- Open Postman application on Ubuntu VM
- Click on "Import" button as shown below



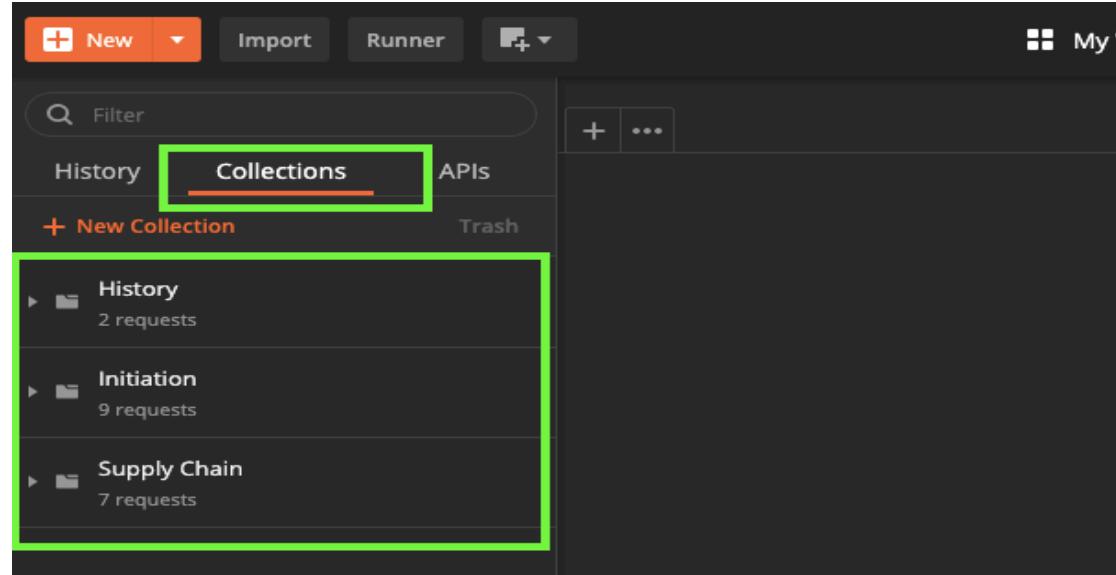
- Click on Choose Files in the pop up that opens



- Navigate to "test" folder available inside project directory in the file explorer window and select all 3 files by pressing "Shift" key and click "Open" as shown



- Click on “Collections” tab where all three collection files will be available as shown below

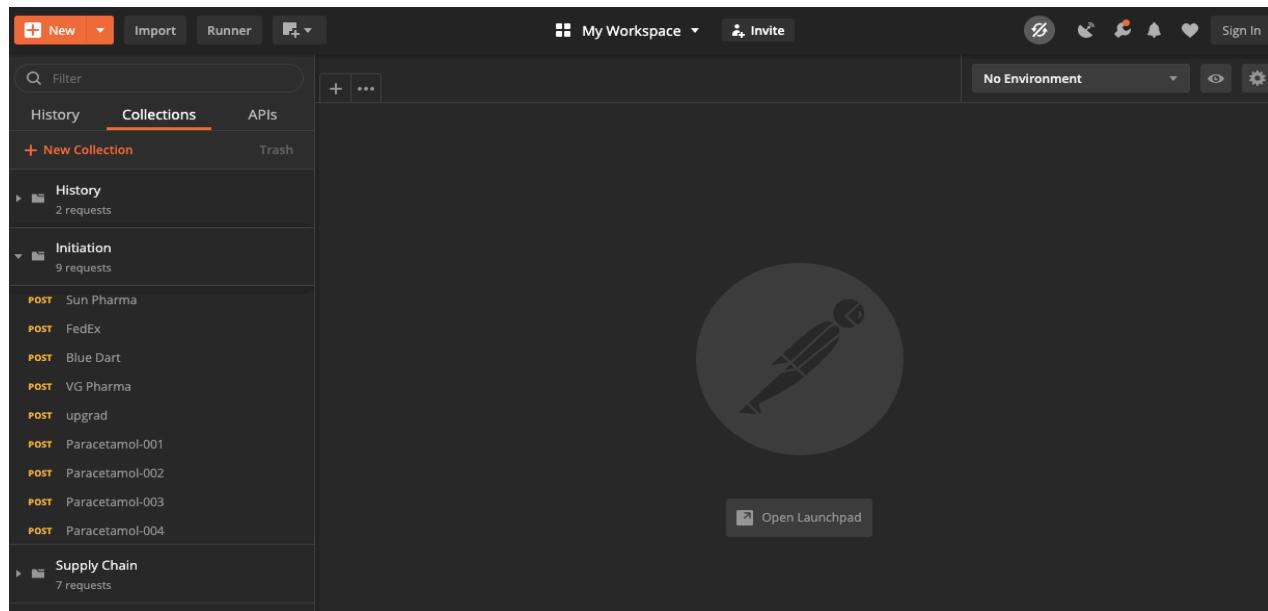


- Before submitting requests from Postman, go to File -> Settings -> General and ensure that “Request timeout in (ms)” is set to 0 (infinite) as shown below to prevent request timeout errors when running smart contracts



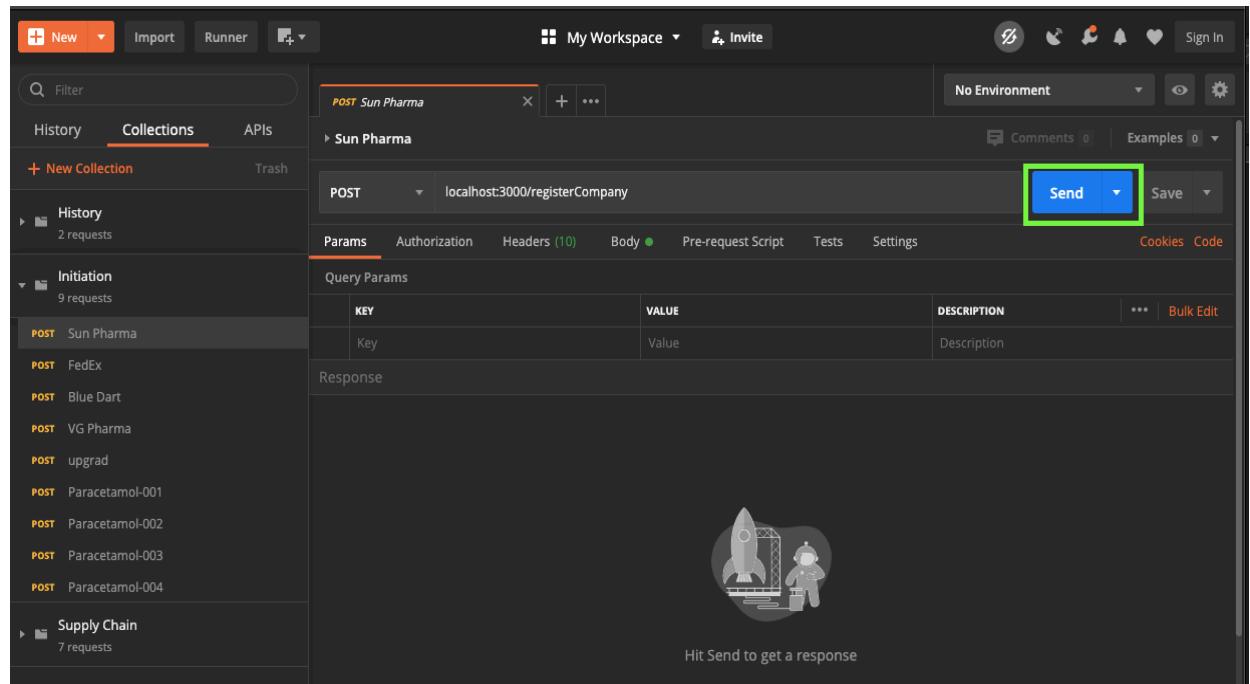
8.2. INITIATION TEST CASES

- This section explains various test cases available as part of "Initiation" collection
- Although all the test cases in a collection can be executed with the click of a single button, **DO NOT EXECUTE INITIATION TEST CASES IN A SINGLE GO**
- Each test case part of this collection will be triggered separately since the requests are submitted to fabric as different organizations
- When first request for an organization is submitted, respective peer of that org builds its own chaincode container image where all dependencies of code are first installed through “npm install” before request is processed since pharma-network is running in Production mode**
- Hence initiation test cases are run one-by-one instead of all at once
- Expand "Initiation" collection by clicking on the small arrow before folder name as shown below



8.2.1. SUN PHARMA

- Click on SV Pharma request to open it in a new tab and click “Send” to submit the request as shown below



The screenshot shows the Postman application interface. On the left, the sidebar lists collections: History, Collections (selected), APIs, and Trash. Under Collections, there is a folder named 'Initiation' containing 9 requests, with 'POST Sun Pharma' selected. Other requests in this folder include POST FedEx, POST Blue Dart, POST VG Pharma, POST upgrad, POST Paracetamol-001, POST Paracetamol-002, POST Paracetamol-003, and POST Paracetamol-004. Below this is another folder named 'Supply Chain' containing 7 requests. The main workspace shows a POST request to 'localhost:3000/registerCompany'. The 'Body' tab is selected, showing a single query parameter 'Key' with value 'Value'. The 'Send' button at the top right of the request card is highlighted with a green box. The status bar at the bottom right says 'Hit Send to get a response'.

- Notice on terminal console output that the node application detects there is no wallet for "Manufacturer" organization and automatically creates it

```
[Manikandan-Ramesh:Application mac$ node .
Distributed Pharma Supply Chain app listening on port 3000!
"/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/manufacturer/MANUFACTURER_ADMIN" does not exist!!!
.....Either wallet does not exist or does not have expected number of files
.....Creating wallet now for Manufacturer
.....Successfully created wallet for Manufacturer
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/manufacturer
.....Connecting to Fabric Gateway
.....Connecting to channel - pharmachannel
.....Connecting to Smart Contract in pharmanet chaincode
.....Register company
.....Processing Register Company Transaction Response

{
  companyID: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
  name: 'Sun Pharma',
  location: 'Chennai',
  organisationRole: 'Manufacturer',
  hierarchyKey: 1
}

.....Register Company Transaction Complete!
.....Disconnecting from Fabric Gateway
New company registered
```

- Since this request is submitted as a Manufacturer, it is sent to peer0 of manufacturer organization for endorsement. As this same peer was used to instantiate chaincode on the channel its corresponding chaincode container is already created and the request completes in few seconds.
- After successful request execution, output is available on Postman under results section

Request details and expected output are explained below

REQUEST

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> organizationName	Manufacturer			
<input checked="" type="checkbox"/> companyCRN	MAN001			
<input checked="" type="checkbox"/> companyName	Sun Pharma			
<input checked="" type="checkbox"/> location	Chennai			
<input checked="" type="checkbox"/> organisationRole	Manufacturer			
Key	Description			

RESPONSE: Expected

The screenshot shows a browser developer tools interface with the 'Body' tab selected. At the top, it displays 'Status: 200 OK' and 'Time: 3.72 s'. Below the status bar, there are tabs for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. The 'JSON' tab is currently active. The main area contains a JSON response with line numbers from 1 to 11. The response object is as follows:

```
1  {
2      "status": "success",
3      "message": "New company registered",
4      "company": {
5          "companyID": "org.pharma-network.pharmanet.companyMAN001Sun Pharma",
6          "name": "Sun Pharma",
7          "location": "Chennai",
8          "organisationRole": "Manufacturer",
9          "hierarchyKey": 1
10     }
11 }
```

8.2.2. FEDEX

- Click on FedEx request to open it in a new tab and click “Send” to submit the request as shown below

The screenshot shows the Postman interface with the following details:

- Request Method: POST
- Request URL: localhost:3000/registerCompany
- Request Headers:
 - Authorization
 - Headers (10)
 - Body (green dot)
 - Pre-request Script
 - Tests
 - Settings
- Query Params:

KEY	VALUE	DESCRIPTION
Key	Value	Description
- Response: (This section is currently empty.)

- Notice on terminal console output that the node application detects there is no wallet for “Transporter” organization and automatically creates it

```
"/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/transporter/TRANSPORTER_ADMIN" does not exist!!!
.....Either wallet does not exist or does not have expected number of files
.....Creating wallet now for Transporter
.....Successfully created wallet for Transporter
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/transporter
.....Connecting to Fabric Gateway
.....Connecting to channel - pharmachannel
.....Connecting to Smart Contract in pharmanet chaincode
.....Register company

.....Processing Register Company Transaction Response

{
  companyID: '\u0000org.pharma-network.pharmanet.company\u0000TRA001\u0000FedEx\u0000',
  name: 'FedEx',
  location: 'Delhi',
  organisationRole: 'Transporter',
  hierarchyKey: null
}

.....Register Company Transaction Complete!
.....Disconnecting from Fabric Gateway
New company registered
```

- Since this is the first request submitted as a Transporter, it is sent to peer0 of transporter organization for endorsement. peer0 of transporter then builds its own chaincode container image where all dependencies of code are first installed through “npm install” before request is processed since pharma-network is running in Production mode. Hence this request can take between 5 to 10 minutes to complete depending on machine configurations and network speed.

- After successful request execution, output is available on Postman under results section

Request details and expected output are explained below

REQUEST :

▶ FedEx

POST localhost:3000/registerCompany

Send Save

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> organizationName	Transporter			
<input checked="" type="checkbox"/> companyCRN	TRA001			
<input checked="" type="checkbox"/> companyName	FedEx			
<input checked="" type="checkbox"/> location	Delhi			
<input checked="" type="checkbox"/> organisationRole	Transporter			
Key	Value	Description		

RESPONSE : Expected

Body Cookies Headers (7) Test Results Status: 200 OK Time: 3 m 32.76 s Size: 485 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2      "status": "success",
3      "message": "New company registered",
4      "company": {
5          "companyID": "org.pharma-network.pharmanet.company@TRA001@FedEx@",
6          "name": "FedEx",
7          "location": "Delhi",
8          "organisationRole": "Transporter",
9          "hierarchyKey": null
10     }
11 }
```

8.2.3. BLUE DART

- Click on Blue Dart request to open it in a new tab and click “Send” to submit the request as shown below

The screenshot shows the Postman application interface. At the top, there's a header bar with 'POST Blue Dart' and a close button. Below the header, the URL 'localhost:3000/registerCompany' is entered. To the right of the URL are 'Send' and 'Save' buttons. The main area has tabs for 'Params', 'Authorization', 'Headers (10)', 'Body (1)', 'Pre-request Script', 'Tests', and 'Settings'. The 'Params' tab is selected. Under 'Query Params', there's a table with one row: 'Key' (empty) and 'Value' (empty). To the right of the table are 'DESCRIPTION' and 'Bulk Edit' buttons. Below the table, there's a section labeled 'Response'.

- Notice on terminal console output that the node application detects wallet is already available for “Transporter” organization (created from previous FedEx request) and does not create it again

```
Creation time of "/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/network/crypto-config/peer0@organizations/transporter.pharma-network.com/users/Admin@transporter.pharma-network.com/msp/" is Fri Jun 19 2020 00:49:57 GMT+0530 (India Standard Time)
Creation time of "/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/transporter/TRANSPORTER_ADMIN" is Fri Jun 19 2020 22:47:11 GMT+0530 (India Standard Time)
.....Wallet exists for Transporter
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/transporter
.....Connecting to Fabric Gateway
.....Connecting to channel - pharmachannel
.....Connecting to Smart Contract in pharmanet chaincode
.....Register company
.....Processing Register Company Transaction Response

{
  companyID: '\u0000org.pharma-network.pharmanet.company\u0000TRA002\u0000Blue Dart\u0000',
  name: 'Blue Dart',
  location: 'Bangalore',
  organisationRole: 'Transporter',
  hierarchyKey: null
}

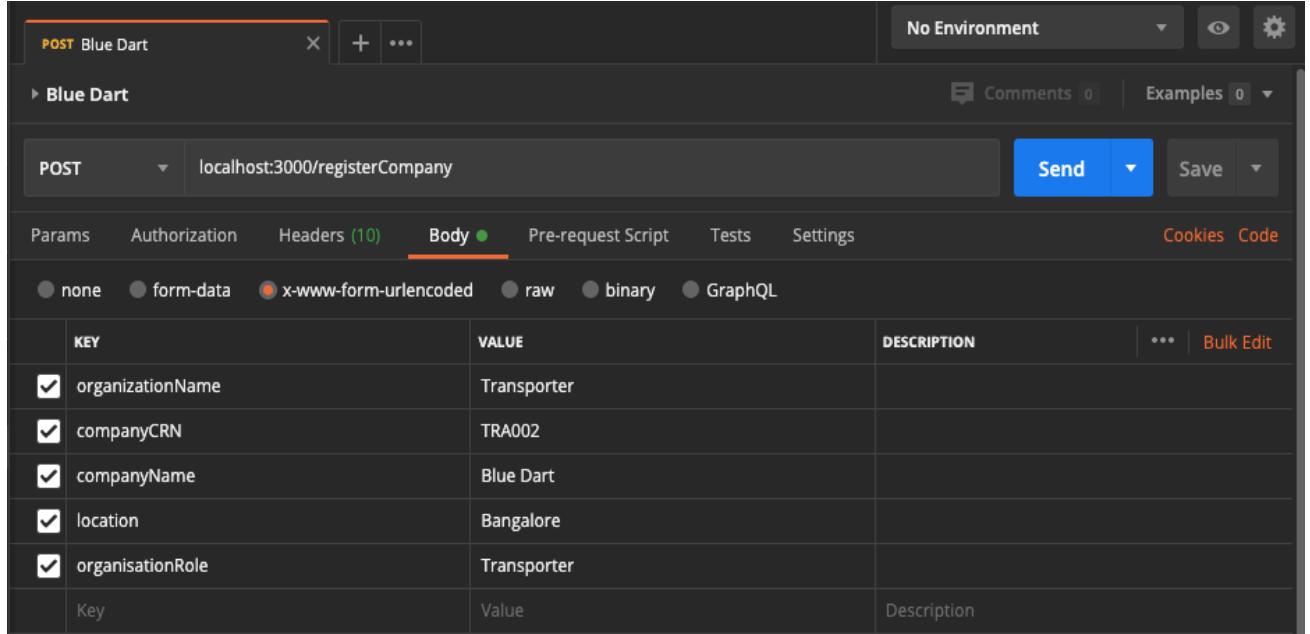
Register Company Transaction Complete!
```

- Since this is the second request submitted as a Transporter which is sent to peer0 of transporter organization for endorsement, its corresponding chaincode container is already created from previous FedEx request and hence this execution completes in few seconds.

- After successful request execution, output is available on Postman under results section

Request details and expected output are explained below

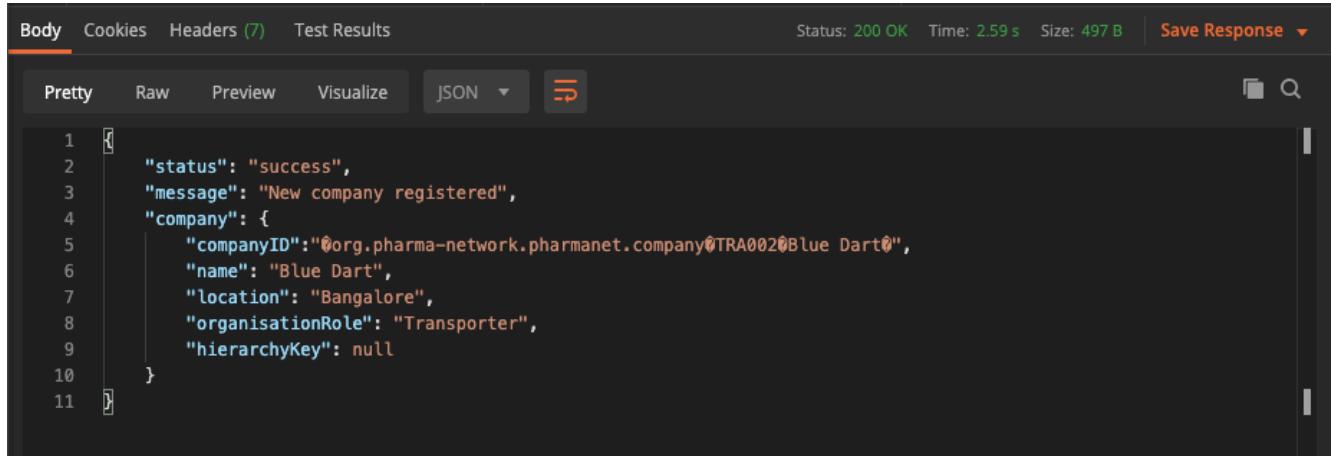
REQUEST :



The screenshot shows the Postman interface for a POST request to 'localhost:3000/registerCompany'. The 'Body' tab is selected, showing form-data parameters:

KEY	VALUE	DESCRIPTION
organizationName	Transporter	
companyCRN	TRA002	
companyName	Blue Dart	
location	Bangalore	
organisationRole	Transporter	
Key	Value	Description

RESPONSE: Expected



The screenshot shows the Postman results panel with a status of 200 OK. The response body is displayed in JSON format:

```

1  {
2   "status": "success",
3   "message": "New company registered",
4   "company": {
5     "companyID": "Org.pharma-network.pharmanet.company@TRA002@Blue Dart@",
6     "name": "Blue Dart",
7     "location": "Bangalore",
8     "organisationRole": "Transporter",
9     "hierarchyKey": null
10   }
11 }
```

8.2.4. VG PHARMA

- Click on VG Pharma request to open it in a new tab and click “Send” to submit the request as shown below

The screenshot shows the Postman application interface. At the top, there is a header bar with the text "POST VG Pharma" and a "No Environment" dropdown. Below the header, the title "VG Pharma" is displayed. On the right side of the header, there are "Comments 0", "Examples 0", and a gear icon. The main workspace shows a POST request to "localhost:3000/registerCompany". The "Params" tab is selected, showing a table for "Query Params" with one row: "Key" and "Value". Below the table, there is a "Response" section. At the bottom of the screen, there are buttons for "Send" (highlighted in blue), "Save", and "Cookies Code".

- Notice on terminal console output that the node application detects there is no wallet for “Distributor” organization and automatically creates it

```
"/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/distributor/DISTRIBUTOR
-ADMIN" does not exist!!!
....Either wallet does not exist or does not have expected number of files
....Creating wallet now for Distributor
....Successfully created wallet for Distributor
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/distributor
....Connecting to Fabric Gateway
....Connecting to channel - pharmachannel
....Connecting to Smart Contract in pharmanet chaincode
....Register company
....Processing Register Company Transaction Response

{
  companyID: '\u0000org.pharma-network.pharmanet.company\u0000DIST001\u0000VG Pharma\u0000',
  name: 'VG Pharma',
  location: 'Vizag',
  organisationRole: 'Distributor',
  hierarchyKey: 2
}

....Register Company Transaction Complete!
....Disconnecting from Fabric Gateway
New company registered
```

- Since this is the first request submitted as a Distributor, it is sent to peer0 of distributor organization for endorsement. peer0 of distributor then builds its own chaincode container image where all dependencies of code are first installed through “npm install” before request is processed since pharma-network is running in Production mode. Hence this request can take between 5 to 10 minutes to complete depending on machine configurations and network speed.

- After successful request execution, output is available on Postman under results section

Request details and expected output are explained below

REQUEST :

The screenshot shows a Postman collection named "VG Pharma". A POST request is selected with the URL "localhost:3000/registerCompany". The "Body" tab is active, showing "x-www-form-urlencoded" selected. The table lists the following parameters:

KEY	VALUE	DESCRIPTION
organizationName	Distributor	
companyCRN	DIST001	
companyName	VG Pharma	
location	Vizag	
organisationRole	Distributor	
Key	Value	Description

RESPONSE: Expected

The screenshot shows the "Test Results" tab of Postman. The response body is displayed in JSON format:

```

1  {
2      "status": "success",
3      "message": "New company registered",
4      "company": {
5          "companyID": "org.pharma-network.pharmanet.company@DIST001@VG Pharma",
6          "name": "VG Pharma",
7          "location": "Vizag",
8          "organisationRole": "Distributor",
9          "hierarchyKey": 2
10     }
11 }

```

Status: 200 OK Time: 3 m 39.90 s Size: 491 B Save Response ▾

8.2.5. UPGRAD

- Click on upgrad request to open it in a new tab and click “Send” to submit the request as shown below

The screenshot shows the Postman interface with a collection named 'upgrad'. A specific POST request is selected, with the URL set to 'localhost:3000/registerCompany'. The 'Params' tab is active, showing a single query parameter 'Key' with a value 'Value'. The 'Send' button is highlighted in blue at the top right of the request panel.

- Notice on terminal console output that the node application detects there is no wallet for “Retailer” organization and automatically creates it

```
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/retailer
.....Connecting to Fabric Gateway
.....Connecting to channel - pharmachannel
.....Connecting to Smart Contract in pharmanet chaincode
.....Register company
.....Processing Register Company Transaction Response

{
  companyID: '\u0000org.pharma-network.pharmanet.company\u0000RET002\u0000upgrad\u0000',
  name: 'upgrad',
  location: 'Mumbai',
  organisationRole: 'Retailer',
  hierarchyKey: 3
}

.....Register Company Transaction Complete!
.....Disconnecting from Fabric Gateway
New company registered
```

- Since this is the first request submitted as a Retailer, it is sent to peer0 of retailer organization for endorsement. peer0 of retailer then builds its own chaincode container image where all dependencies of code are first installed through “npm install” before request is processed since pharma-network is running in Production mode. Hence this request can take between 5 to 10 minutes to complete depending on machine configurations and network speed.

- After successful request execution, output is available on Postman under results section

Request details and expected output are explained below

REQUEST:

The screenshot shows the Postman application interface. At the top, it displays a POST request titled "upgrad" with the URL "localhost:3000/registerCompany". The "Body" tab is selected, showing a form-data structure with the following fields:

KEY	VALUE	DESCRIPTION
organizationName	Retailer	
companyCRN	RET002	
companyName	upgrad	
location	Mumbai	
organisationRole	Retailer	
Key	Value	Description

Below the table, there are buttons for "Send" and "Save". The "Headers" tab shows 10 headers. The "Tests" and "Settings" tabs are also visible.

RESPONSE: Expected

POST upgrad

No Environment

POST localhost:3000/registerCompany Send Save

Body Cookies Headers (7) Test Results Status: 200 OK Time: 3 m 43.87 s Size: 482 B Save Response

Pretty Raw Preview Visualize JSON

```
1   {
2     "status": "success",
3     "message": "New company registered",
4     "company": {
5       "companyID": "org.pharma-network.pharmanet.company@RET002@upgrad@",
6       "name": "upgrad",
7       "location": "Mumbai",
8       "organisationRole": "Retailer",
9       "hierarchyKey": 3
10    }
11 }
```

8.2.6. PARACETAMOL-001 TO PARACETAMOL-004

- There are 4 requests under “Initiation” collection that are used to create four drug entities with name “Paracetamol” and serial numbers 001 to 004 as shown below

The screenshot shows the Hyperledger Fabric console interface. A tree view on the left has 'Initiation' expanded, showing '9 requests'. Below this, a list of requests is displayed with the following details:

- POST Sun Pharma**
- POST FedEx**
- POST Blue Dart**
- POST VG Pharma**
- POST upgrad**
- POST Paracetamol-001**
- POST Paracetamol-002**
- POST Paracetamol-003**
- POST Paracetamol-004**

The last four requests (Paracetamol-001 to Paracetamol-004) are highlighted with a green rectangular border.

- Open each of these requests individually and click on Send button to submit the requests
- Notice on terminal console output that the node application detects wallet is available for “Manufacturer” organization and does not create it again for any of these requests

```
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/manufacturer
....Connecting to Fabric Gateway
....Connecting to channel - pharmachannel
....Connecting to Smart Contract in pharmanet chaincode
....Add drug
....Processing Add Drug Transaction Response

{
  productID: '\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u0000001\u0000',
  name: 'Paracetamol',
  manufacturer: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
  manufacturingDate: '2020-06-19T00:00:00Z',
  expiryDate: '2022-09-18T00:00:00Z',
  owner: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
  shipment: [],
  createdat: '2020-06-19T18:12:14.276Z',
  updatedat: '2020-06-19T18:12:14.276Z'
}

....Add Drug Transaction Complete!
....Disconnecting from Fabric Gateway
New drug added
Creation time of "/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/network/crypto-config/peerOrganizations/manufacturer.pharma-network.com/users/Admin@manufacturer.pharma-network.com/msp/" is Fri Jun 19 2020 20:49:56 GMT+0530 (India Standard Time)
Creation time of "/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/manufacturer/MANUFACTURER_ADMIN" is Fri Jun 19 2020 22:41:35 GMT+0530 (India Standard Time)
....Wallet exists for Manufacturer
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/manufacturer
....Connecting to Fabric Gateway
....Connecting to channel - pharmachannel
....Connecting to Smart Contract in pharmanet chaincode
....Add drug
....Processing Add Drug Transaction Response

{
  productID: '\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u0000002\u0000',
  name: 'Paracetamol',
  manufacturer: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
  manufacturingDate: '2020-06-19T00:00:00Z',
  expiryDate: '2022-09-18T00:00:00Z',
  owner: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
  shipment: [],
  createdat: '2020-06-19T18:12:22.510Z',
  updatedat: '2020-06-19T18:12:22.510Z'
}

....Add Drug Transaction Complete!
....Disconnecting from Fabric Gateway
New drug added
```

```

Creation time of "/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/network/crypto-config/peerOrganizations/manufacturer.pharma-network.com/users/Admin@manufacturer.pharma-network.com/msp/" is Fri Jun 19 2020 20:49:57 GMT+0530 (India Standard Time)
Creation time of "/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/manufacturer/MANUFACTURER_ADMIN" is Fri Jun 19 2020 22:41:35 GMT+0530 (India Standard Time)
.....Wallet exists for Manufacturer
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/manufacturer
.....Connecting to Fabric Gateway
.....Connecting to channel - pharmachannel
.....Connecting to Smart Contract in pharmanet chaincode
.....Add drug
.....Processing Add Drug Transaction Response

{

productID: '\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u0000003\u0000',
name: 'Paracetamol',
manufacturer: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
manufacturingDate: '2020-06-19T00:00:00.000Z',
expiryDate: '2022-09-18T00:00:00.000Z',
owner: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
shipment: [],
createdAt: '2020-06-19T18:12:30.177Z',
updatedAt: '2020-06-19T18:12:30.177Z'
}

....Add Drug Transaction Complete!
....Disconnecting from Fabric Gateway
New drug added
Creation time of "/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/network/crypto-config/peerOrganizations/manufacturer.pharma-network.com/users/Admin@manufacturer.pharma-network.com/msp/" is Fri Jun 19 2020 20:49:57 GMT+0530 (India Standard Time)
Creation time of "/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/manufacturer/MANUFACTURER_ADMIN" is Fri Jun 19 2020 22:41:35 GMT+0530 (India Standard Time)
.....Wallet exists for Manufacturer
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/manufacturer
.....Connecting to Fabric Gateway
.....Connecting to channel - pharmachannel
.....Connecting to Smart Contract in pharmanet chaincode
.....Add drug
.....Processing Add Drug Transaction Response

{

productID: '\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u0000004\u0000',
name: 'Paracetamol',
manufacturer: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
manufacturingDate: '2020-06-19T00:00:00.000Z',
expiryDate: '2022-09-18T00:00:00.000Z',
owner: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
shipment: [],
createdAt: '2020-06-19T18:12:37.770Z',
updatedAt: '2020-06-19T18:12:37.770Z'
}

```

- Since these requests are submitted as Manufacturer, each request is sent to peer0 of manufacturer organization for endorsement and the requests complete in few seconds since chaincode container is already created as explained above
- After successful execution, output is available on Postman under results section of respective request

Request details and expected output are shown below

8.2.6.1. PARACETAMOL-001

REQUEST:

The screenshot shows a Postman collection named "Paracetamol-001". The active request is a POST to "localhost:3000/addDrug". The "Body" tab is selected, showing a form-data payload with the following fields:

KEY	VALUE	DESCRIPTION
organizationName	Manufacturer	
drugName	Paracetamol	
serialNo	001	
mfgDate	2020-06-19	
expDate	2022-09-18	
companyCRN	MAN001	
Key	Value	Description

RESPONSE: Expected

The screenshot shows the "Body" tab of the Postman response. The status is 200 OK. The response is a JSON object:

```

1  [
2      "status": "success",
3      "message": "New drug added",
4      "drug": {
5          "productID": "org.pharma-network.pharmanet.drug@Paracetamol@001",
6          "name": "Paracetamol",
7          "manufacturer": "org.pharma-network.pharmanet.company@MAN001@Sun Pharma",
8          "manufacturingDate": "2020-06-19T00:00:00.000Z",
9          "expiryDate": "2022-09-18T00:00:00.000Z",
10         "owner": "org.pharma-network.pharmanet.company@MAN001@Sun Pharma",
11         "shipment": [],
12         "createdAt": "2020-06-19T18:12:14.276Z",
13         "updatedAt": "2020-06-19T18:12:14.276Z"
14     }
15 ]

```

8.2.6.2. PARACETAMOL-002

REQUEST:

The screenshot shows a Postman interface with the following details:

- Request Method:** POST
- URL:** localhost:3000/addDrug
- Body (x-www-form-urlencoded):**
 - organizationName: Manufacturer
 - drugName: Paracetamol
 - serialNo: 002
 - mfgDate: 2020-06-19
 - expDate: 2022-09-18
 - companyCRN: MAN001
- Headers (10):** (This section is visible but empty in the screenshot)
- Tests:** (This section is visible but empty in the screenshot)
- Settings:** (This section is visible but empty in the screenshot)
- Cookies:** (This section is visible but empty in the screenshot)
- Code:** (This section is visible but empty in the screenshot)

RESPONSE: Expected

The screenshot shows the Postman interface with the following details:

- Body:** (Pretty) JSON response
- Status:** 200 OK
- Time:** 2.45 s
- Size:** 769 B
- Response Content:**

```
1  {
2      "status": "success",
3      "message": "New drug added",
4      "drug": {
5          "productID": "org.pharma-network.pharmanet.drug@Paracetamol@002@",
6          "name": "Paracetamol",
7          "manufacturer": "org.pharma-network.pharmanet.company@MAN001@Sun Pharma",
8          "manufacturingDate": "2020-06-19T00:00:00.000Z",
9          "expiryDate": "2022-09-18T00:00:00.000Z",
10         "owner": "org.pharma-network.pharmanet.company@MAN001@Sun Pharma",
11         "shipment": [],
12         "createdAt": "2020-06-19T18:12:22.510Z",
13         "updatedAt": "2020-06-19T18:12:22.510Z"
14     }
15 }
```

8.2.6.3. PARACETAMOL-003

REQUEST:

The screenshot shows the Postman interface with a POST request to `localhost:3000/addDrug`. The 'Body' tab is selected, showing the following parameters:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> organizationName	Manufacturer	
<input checked="" type="checkbox"/> drugName	Paracetamol	
<input checked="" type="checkbox"/> serialNo	003	
<input checked="" type="checkbox"/> mfgDate	2020-06-19	
<input checked="" type="checkbox"/> expDate	2022-09-18	
<input checked="" type="checkbox"/> companyCRN	MAN001	
Key	Value	Description

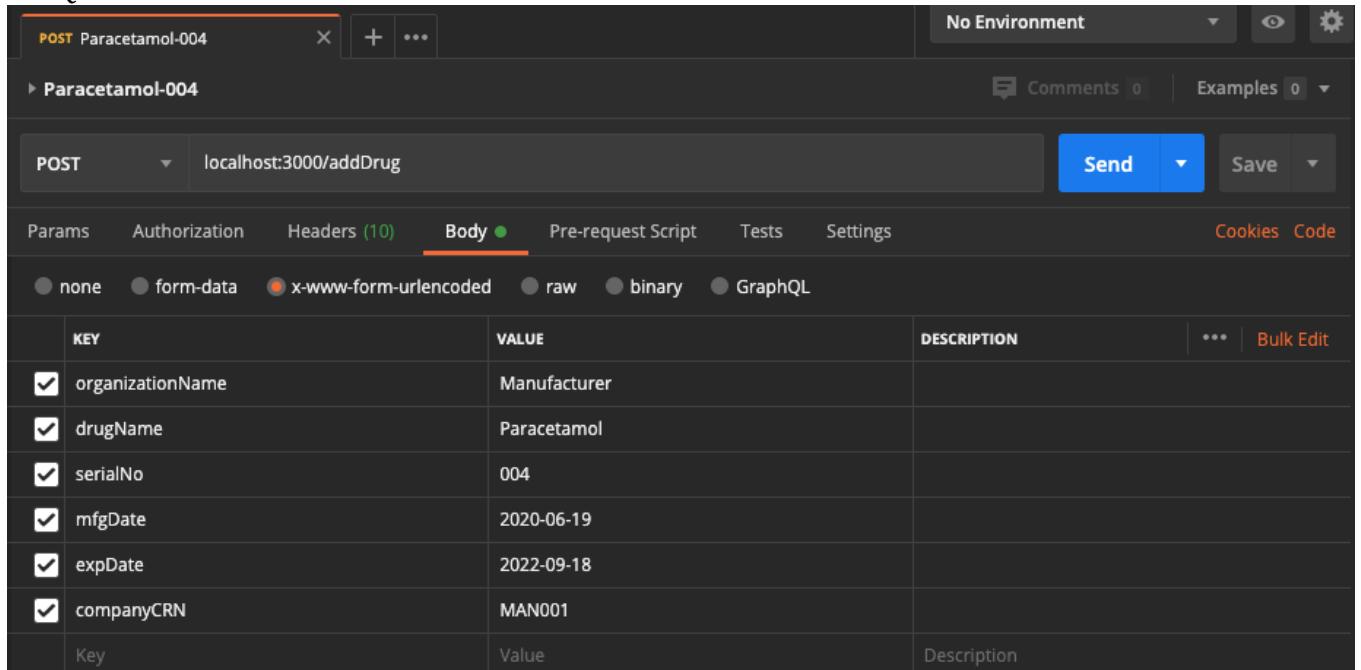
RESPONSE: Expected

The screenshot shows the Postman interface displaying the expected JSON response for the drug addition. The response is:

```
1 [ {  
2     "status": "success",  
3     "message": "New drug added",  
4     "drug": {  
5         "productID": "org.pharma-network.pharmanet.drug@Paracetamol@003@",  
6         "name": "Paracetamol",  
7         "manufacturer": "org.pharma-network.pharmanet.company@MAN001@Sun Pharma@",  
8         "manufacturingDate": "2020-06-19T00:00:00.000Z",  
9         "expiryDate": "2022-09-18T00:00:00.000Z",  
10        "owner": "org.pharma-network.pharmanet.company@MAN001@Sun Pharma@",  
11        "shipment": [],  
12        "createdAt": "2020-06-19T18:12:30.177Z",  
13        "updatedAt": "2020-06-19T18:12:30.177Z"  
14    }  
15 }
```

8.2.6.4. PARACETAMOL-004

REQUEST:

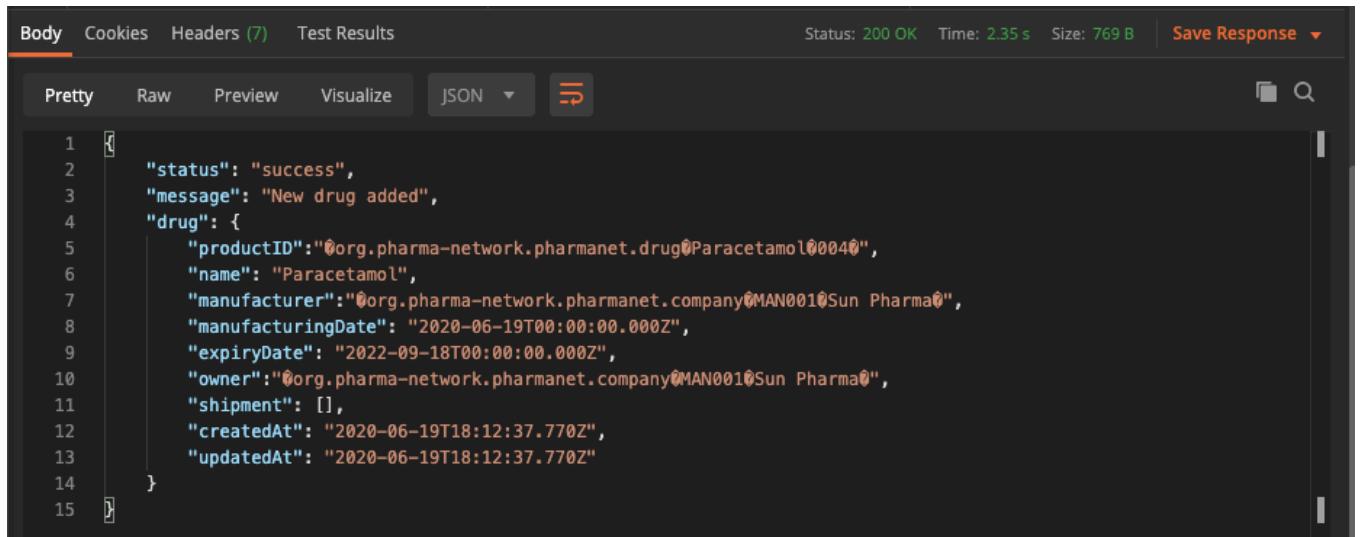


The screenshot shows a Postman interface with the following details:

- Request URL:** POST Paracetamol-004
- Method:** POST
- Endpoint:** localhost:3000/addDrug
- Body Type:** x-www-form-urlencoded
- Body Parameters:**

KEY	VALUE	DESCRIPTION
organizationName	Manufacturer	
drugName	Paracetamol	
serialNo	004	
mfgDate	2020-06-19	
expDate	2022-09-18	
companyCRN	MAN001	
Key	Value	Description
- Environment:** No Environment
- Comments:** 0
- Examples:** 0

RESPONSE : Expected



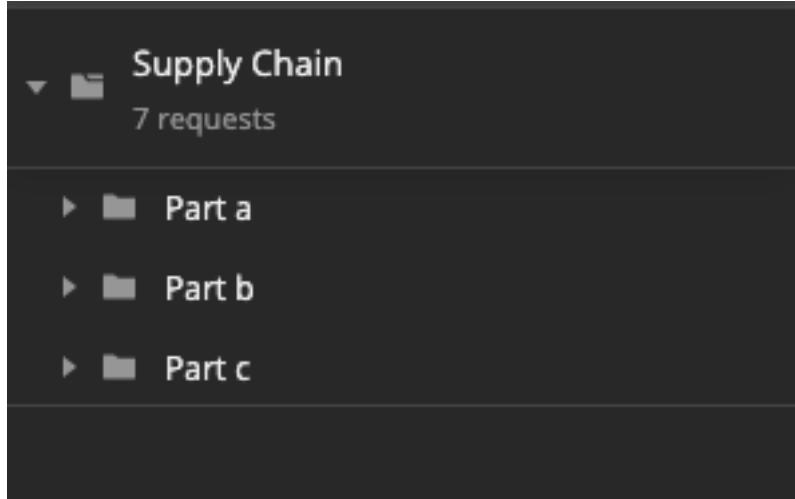
The screenshot shows the JSON response body from the previous request:

```
1 {
2     "status": "success",
3     "message": "New drug added",
4     "drug": {
5         "productID": "org.pharma-network.pharmanet.drug@Paracetamol@004",
6         "name": "Paracetamol",
7         "manufacturer": "org.pharma-network.pharmanet.company@MAN001@Sun Pharma",
8         "manufacturingDate": "2020-06-19T00:00:00.000Z",
9         "expiryDate": "2022-09-18T00:00:00.000Z",
10        "owner": "org.pharma-network.pharmanet.company@MAN001@Sun Pharma",
11        "shipment": [],
12        "createdAt": "2020-06-19T18:12:37.770Z",
13        "updatedAt": "2020-06-19T18:12:37.770Z"
14    }
15 }
```

The response indicates a successful operation, adding a new drug named Paracetamol with serial number 004, manufactured by Sun Pharma, and an expiry date of September 18, 2022.

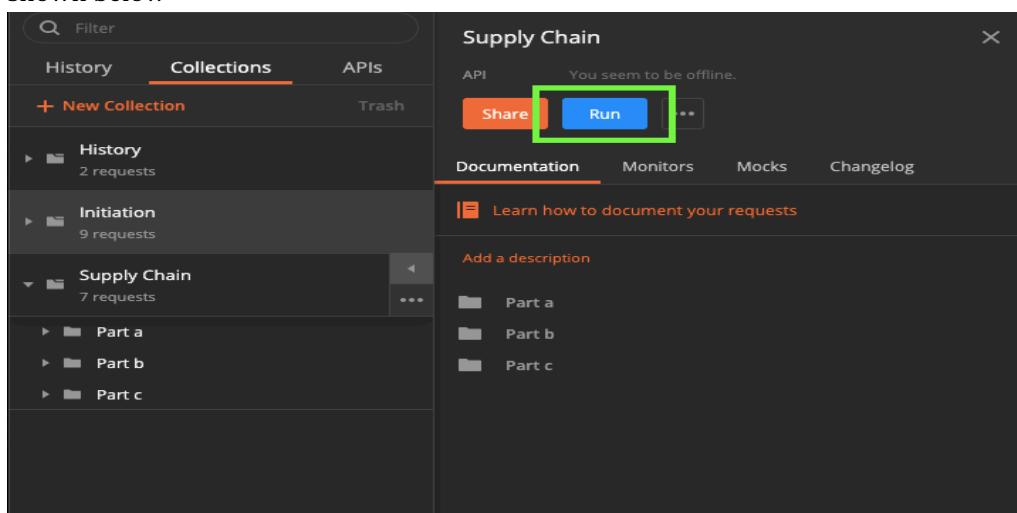
8.3. SUPPLY CHAIN TEST CASES

- This section explains various test cases available as part of “Supply Chain” collection
- The collection is divided into three parts grouped under three folders as shown below



□ These collections can be triggered at once with appropriate time delay between each request since they are submitted as orgs that have peer chaincode containers already created from Initiation section as explained above and hence complete in few seconds

- Wallets for all these orgs are also created from Initiation section and hence are not re-created as explained above
- Click on arrow button on Supply Chain collection and in the fly out menu click “Run” as shown below



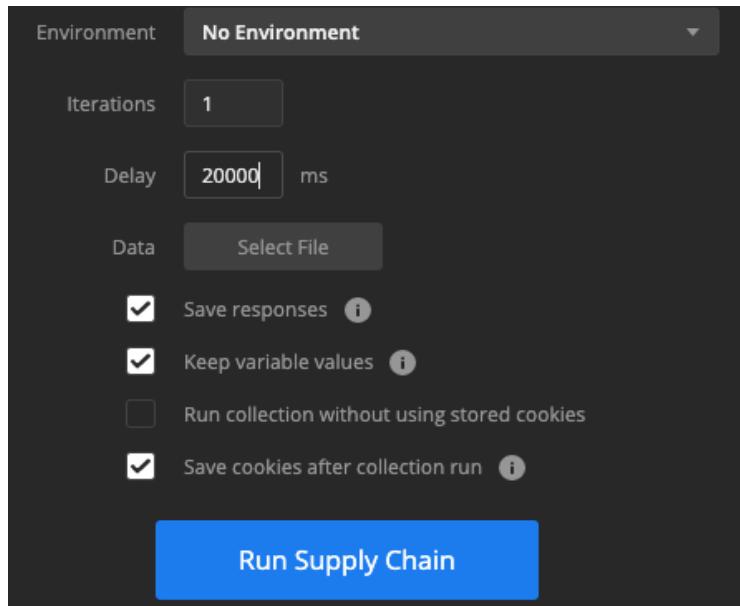
- This will launch Postman collection runner in a separate window as shown below

The screenshot shows the 'Collection Runner' window in 'My Workspace'. On the left, there's a sidebar with a search bar and a tree view of collections: 'Supply Chain' (selected), 'Part a', 'Part b', and 'Part c'. On the right, under 'RUN ORDER', there's a list of selected requests: 'VGPharma-PO', 'Sun Pharma-Shipment', 'FedEx-DeliverShipment', 'upgrad-PO', 'VG Pharma-Shipment', 'Blue Dart-Deliver Shipment', and 'upgrad-Retail Drug'. Below the sidebar are configuration options: 'Environment' set to 'No Environment', 'Iterations' set to 1, 'Delay' set to 0 ms, 'Data' with a 'Select File' button, and several checkboxes: 'Save responses' (unchecked), 'Keep variable values' (checked), 'Run collection without using stored cookies' (unchecked), and 'Save cookies after collection run' (checked). At the bottom is a large blue button labeled 'Run Supply Chain'.

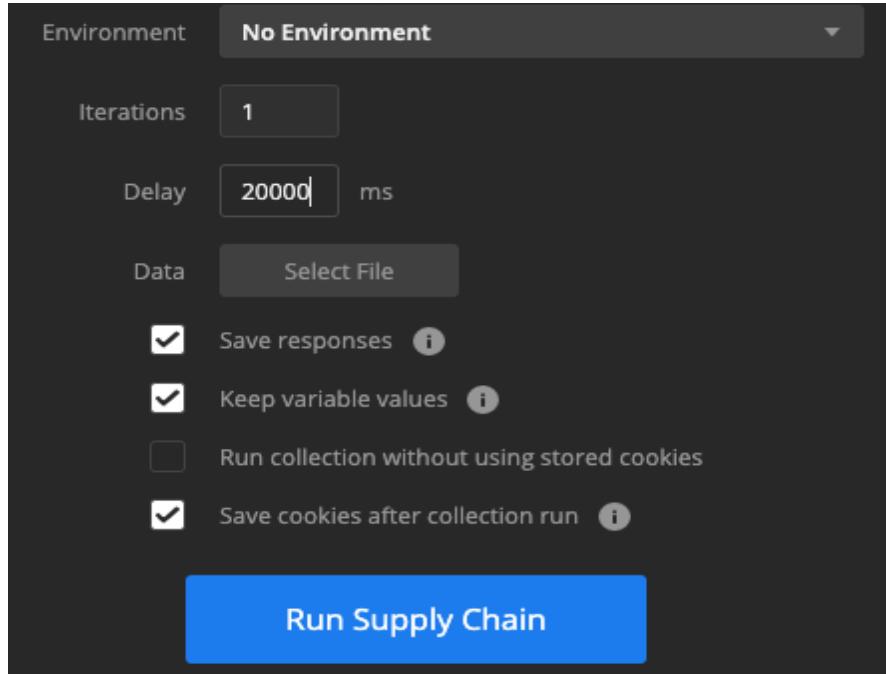
- Each folder in the collection will be executed separately from collection runner

8.3.1. PART A

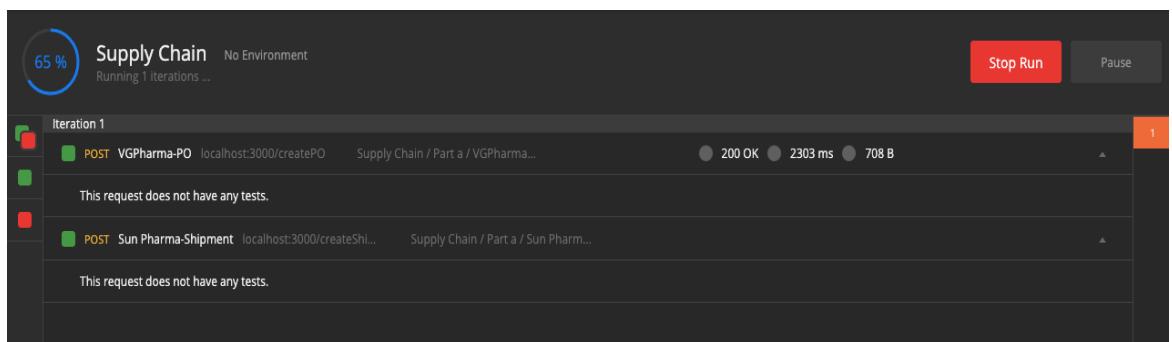
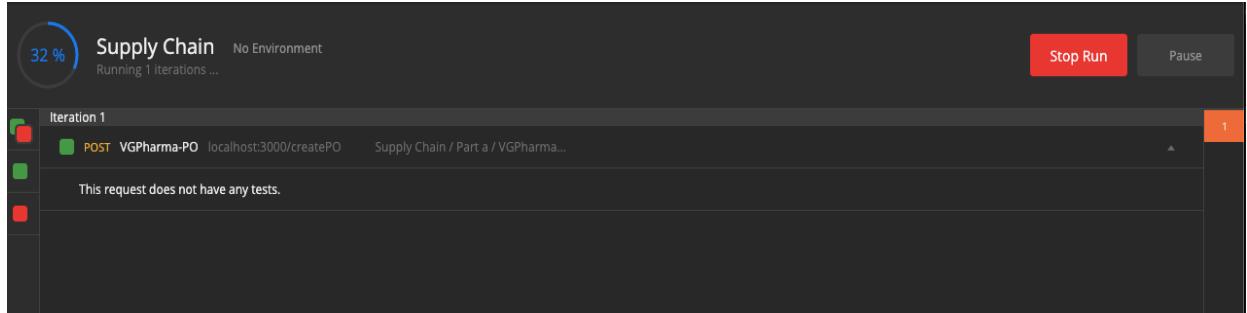
- Click on Part a folder on Collection Runner window
- Ensure settings are as shown below and note that delay between requests is set at 20000 ms which is 20 seconds
- Click on Save response to store the response while running a collection



- Scroll down and click “Run Supply Chain” as shown below



- This will open Run Results tab where the run progress can be monitored



- Once the run completes, screen will look like below

Collection Runner Run Results My Workspace ▾ Run In Command Line Docs

Supply Chain No Environment a min ago

Iteration 1

- POST VGPharma-PO localhost:3000/createPO Supply Chain / Part a / VGPharma... 200 OK 3495 ms 708 B
- POST Sun Pharma-Shipment localhost:3000/createShi... Supply Chain / Part a / Sun Pharm... 200 OK 2356 ms 917 B
- PUT FedEx-DeliveryShipment localhost:3000/updateSh... Supply Chain / Part a / FedEx-Deli... 200 OK 2425 ms 1.957 KB

- The request parameters and response body for each request can be viewed by clicking on method name in Run Results tab as shown below

Collection Runner Run Results My Workspace ▾ Run In Command Line Docs

Supply Chain No Environment a min ago

Iteration 1

- POST VGPharma-PO localhost:3000/createPO Supply Chain / Part a / VGPharma... 200 OK

This request does not have any tests.

Request URL
Request Headers (9)
Request Body
Response Headers
Response Body

8.3.1.1. VGPHARMA-PO

REQUEST:

Collection Runner Run Results My Workspace ▾ Run In Command Line Docs

Supply Chain No Environment a min ago

Iteration 1

- POST VGPharma-PO localhost:3000/createPO Supply Chain / Part a / VGPharma... 200 OK

This request does not have any tests.

Request URL
Request Headers (9)
Request Body

```
{"organizationName": "Distributor", "buyerCRN": "DIST001", "sellerCRN": "MAN001", "drugName": "Paracetamol", "quantity": "3"}
```

This request does not have any tests.

RESPONSE: Expected

The screenshot shows a REST API testing interface with the following details:

- Iteration 1**
- POST VGPharma-PO** to `localhost:3000/createPO` under `Supply Chain / Part a / VGPharma...`
- Response Headers (7)**: This section is collapsed.
- Response Body**: This section is expanded, showing the following JSON response:

```
{  
  "status": "success",  
  "message": "New purchase order created",  
  "purchaseorder": {  
    "poID": "\u0000org.pharma-",  
    "drugName": "Paracetamol",  
    "quantity": 3,  
    "buyer": "\u0000org.pharma-",  
    "seller": "\u0000org.pharma-",  
    "createdAt": "2020-06-19T18:58:52.207Z",  
    "updatedAt": "2020-06-19T18:58:52.207Z"  
  }  
}
```

8.3.1.2. SUN PHARMA-SHIPMENT

REQUEST:

The screenshot shows a REST API testing interface with the following details:

- Iteration 1**
- POST VGPharma-PO** to `localhost:3000/createPO` under `Supply Chain / Part a / VGPharma...`
- Response Headers (7)**: This section is collapsed.
- Response Body**: This section is expanded, showing the following JSON response:

```
{  
  "status": "success",  
  "message": "New purchase order created",  
  "purchaseorder": {  
    "poID": "\u0000org.pharma-",  
    "drugName": "Paracetamol",  
    "quantity": 3,  
    "buyer": "\u0000org.pharma-",  
    "seller": "\u0000org.pharma-",  
    "createdAt": "2020-06-19T18:58:52.207Z",  
    "updatedAt": "2020-06-19T18:58:52.207Z"  
  }  
}
```

RESPONSE: Expected

Iteration 1

POST VGPharma-PO localhost:3000/createPO Supply Chain / Part a / VGPharma...

This request does not have any tests.

POST Sun Pharma-Shipment localhost:3000/createShi... Supply Chain / Part a / Sun Pharm...

This request does not have any tests.

PUT F Response Body

PUT F Response Body

```
{  
    "status": "success",  
    "message": "New shipment created",  
    "shipment": {  
        "shipmentID": "\u0000org.pharma-  
network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u0  
000",  
        "creator": "\u0000org.pharma-  
network.pharmanet.company\u0000MAN001\u0000Sun  
Pharma\u0000",  
        "assets": ["\u0000org.pharma-  
network.pharmanet.drug\u0000Paracetamol\u0000001\u0000",  
"\u0000org.pharma-  
network.pharmanet.drug\u0000Paracetamol\u0000002\u0000",  
"\u0000org.pharma-  
network.pharmanet.drug\u0000Paracetamol\u0000003\u0000"],  
        "transporter": "\u0000org.pharma-  
network.pharmanet.company\u0000TRA001\u0000FedEx\u0000",  
        "status": "in-transit",  
        "createdAt": "2020-06-19T18:59:14.504Z",  
        "updatedAt": "2020-06-19T18:59:14.504Z"  
    }  
}
```

8.3.1.3. FEDEX-DELIVER SHIPMENT

REQUEST:

Iteration 1

POST VGPharma-PO localhost:3000/createPO Supply Chain / Part a / VGPharma...

This request does not have any tests.

POST Sun Pharma-Shipment localhost:3000/createShi... Supply Chain / Part a / Sun Pharm...

This request does not have any tests.

PUT FedEx-DeliverShipment localhost:3000/updateSh... Supply Chain / Part a / FedEx-Deli...

This request does not have any tests.

Request URL

Request Headers (9)

Request Body

```
{"organizationName": "Transporter", "buyerCRN": "DIST001", "drugName": "Paracetamol", "transporterCRN": "TRA001"}
```

Response Headers (7)

Response Body

RESPONSE : Expected

Iteration 1

	POST	VGPharma-PO	localhost:3000/createPO	Supply Chain / Part a / VGPharma...
This request does not have any tests.				
	POST	Sun Pharma-Shipment	localhost:3000/createShi...	Supply Chain / Part a / Sun Pharm...
This request does not have any tests.				
	PUT	FedEx-DeliverShipment	localhost:3000/updateSh...	Supply Chain / Part a / FedEx-Deli...
This request does not have any tests.				

```
{
  "status": "success",
  "message": "Shipment updated",
  "assets": [
    {
      "productID": "\u0000org.pharma-
network.pharmanet.drug\u0000Paracetamol\u0000001\u0000",
      "name": "Paracetamol",
      "manufacturer": "\u0000org.pharma-
network.pharmanet.company\u0000MAN001\u0000Sun
Pharma\u0000",
      "manufacturingDate": "2020-06-19T00:00:00.000Z",
      "expiryDate": "2022-09-18T00:00:00.000Z",
      "owner": "\u0000org.pharma-
network.pharmanet.company\u0000DIST001\u0000VG
Pharma\u0000",
      "shipment": ["\u0000org.pharma-
network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u00
00"],
      "createdAt": "2020-06-19T18:55:45.970Z",
      "updatedAt": "2020-06-19T18:59:36.849Z"
    },
    {
      "productID": "\u0000org.pharma-
network.pharmanet.drug\u0000Paracetamol\u0000002\u0000",
      "name": "Paracetamol",
      "manufacturer": "\u0000org.pharma-
network.pharmanet.company\u0000MAN002\u0000Sun
Pharma\u0000",
      "manufacturingDate": "2020-06-19T00:00:00.000Z",
      "expiryDate": "2022-09-18T00:00:00.000Z",
      "owner": "\u0000org.pharma-
network.pharmanet.company\u0000DIST001\u0000VG
Pharma\u0000",
      "shipment": ["\u0000org.pharma-
network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u00
00"],
      "createdAt": "2020-06-19T18:55:50.582Z",
      "updatedAt": "2020-06-19T18:59:36.853Z"
    }
  ],
  "owner": "\u0000org.pharma-
network.pharmanet.company\u0000MAN001\u0000Sun
Pharma\u0000",
  "manufacturingDate": "2020-06-19T00:00:00.000Z",
  "expiryDate": "2022-09-18T00:00:00.000Z",
  "owner": "\u0000org.pharma-
network.pharmanet.company\u0000DIST001\u0000VG
Pharma\u0000",
  "shipment": ["\u0000org.pharma-
network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u00
00"]
}
```

Iteration 1

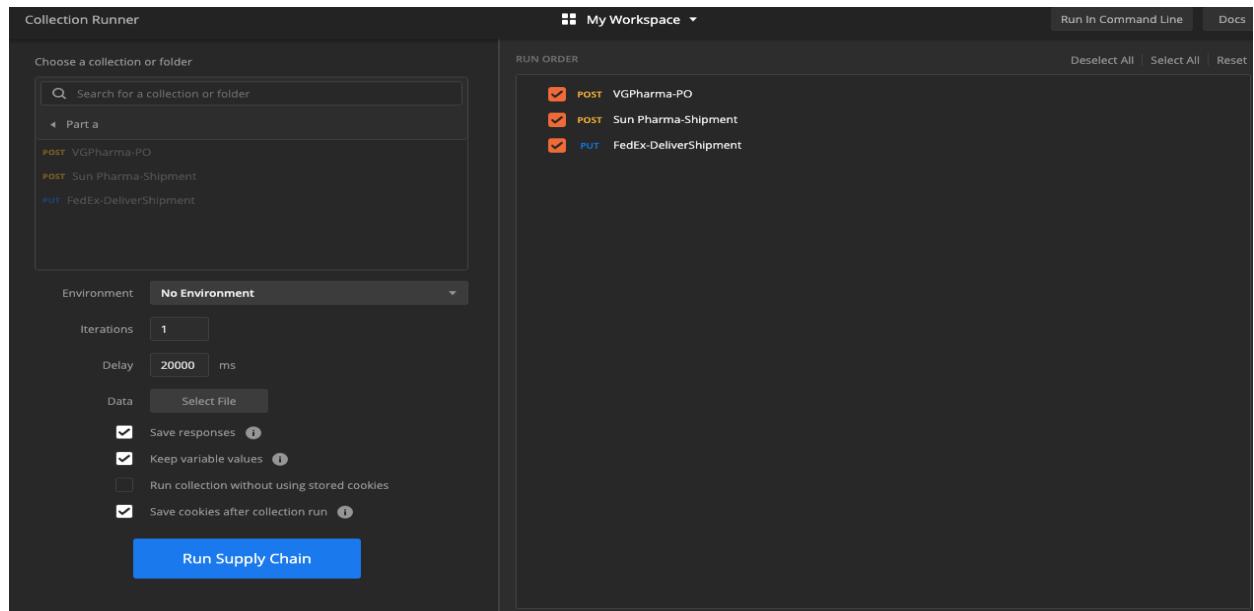
	POST	VGPharma-PO	localhost:3000/createPO	Supply Chain / Part a / VGPharma...
This request does not have any tests.				
	POST	Sun Pharma-Shipment	localhost:3000/createShi...	Supply Chain / Part a / Sun Pharm...
This request does not have any tests.				
	PUT	FedEx-DeliverShipment	localhost:3000/updateSh...	Supply Chain / Part a / FedEx-Deli...
This request does not have any tests.				

```
{
  "status": "success",
  "message": "Shipment updated",
  "assets": [
    {
      "productID": "\u0000org.pharma-
network.pharmanet.company\u0000MAN001\u0000Sun
Pharma\u0000",
      "name": "Paracetamol",
      "manufacturer": "\u0000org.pharma-
network.pharmanet.company\u0000MAN001\u0000Sun
Pharma\u0000",
      "manufacturingDate": "2020-06-19T00:00:00.000Z",
      "expiryDate": "2022-09-18T00:00:00.000Z",
      "owner": "\u0000org.pharma-
network.pharmanet.company\u0000DIST001\u0000VG
Pharma\u0000",
      "shipment": ["\u0000org.pharma-
network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u00
00"],
      "createdAt": "2020-06-19T18:55:45.970Z",
      "updatedAt": "2020-06-19T18:59:36.849Z"
    },
    {
      "productID": "\u0000org.pharma-
network.pharmanet.drug\u0000Paracetamol\u0000003\u0000",
      "name": "Paracetamol",
      "manufacturer": "\u0000org.pharma-
network.pharmanet.company\u0000MAN001\u0000Sun
Pharma\u0000",
      "manufacturingDate": "2020-06-19T00:00:00.000Z",
      "expiryDate": "2022-09-18T00:00:00.000Z",
      "owner": "\u0000org.pharma-
network.pharmanet.company\u0000DIST001\u0000VG
Pharma\u0000",
      "shipment": ["\u0000org.pharma-
network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u00
00"]
    }
  ],
  "owner": "\u0000org.pharma-
network.pharmanet.company\u0000MAN001\u0000Sun
Pharma\u0000",
  "manufacturingDate": "2020-06-19T00:00:00.000Z",
  "expiryDate": "2022-09-18T00:00:00.000Z",
  "owner": "\u0000org.pharma-
network.pharmanet.company\u0000DIST001\u0000VG
Pharma\u0000",
  "shipment": ["\u0000org.pharma-
network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u00
00"]
}
```

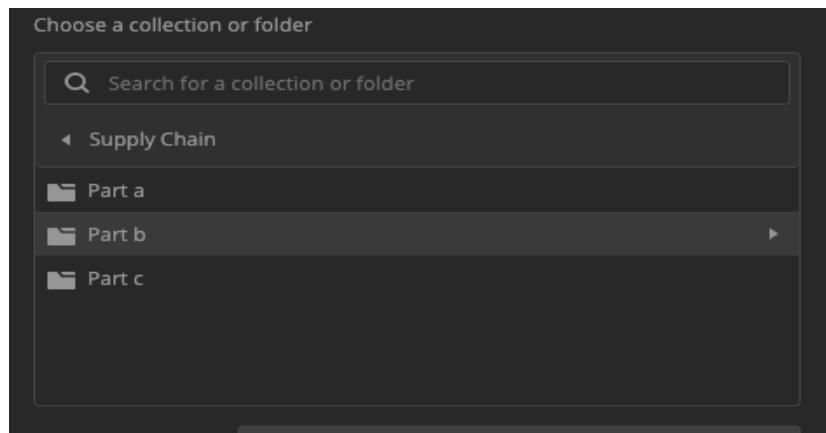
Iteration 1			
█	POST	VGPharma-PO	localhost:3000/createPO Supply Chain / Part a / VGPharma...
█	This request does not have any tests.		
█	POST	Sun Pharma-Shipment	localhost:3000/createShi... Supply Chain / Part a / Sun Pharm...
█	This request does not have any tests.		
█	PUT	FedEx-DeliverShipment	localhost:3000/updateSh... Supply Chain / Part a / FedEx-Deli...
This request does not have any tests.			
<pre> "shipment": ["\u0000org.pharma- network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u00 000"], "createdAt": "2020-06-19T18:55:50.582Z", "updatedAt": "2020-06-19T18:59:36.853Z" }, { "productID": "\u0000org.pharma- network.pharmanet.drug\u0000Paracetamol\u0000003\u0000", "name": "Paracetamol", "manufacturer": "\u0000org.pharma- network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharma- network.pharmanet.company\u0000DIST001\u0000VG Pharma\u0000", "shipment": ["\u0000org.pharma- network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u00 000"], "createdAt": "2020-06-19T18:55:55.400Z", "updatedAt": "2020-06-19T18:59:36.855Z" }] </pre>			

8.3.2. PART B

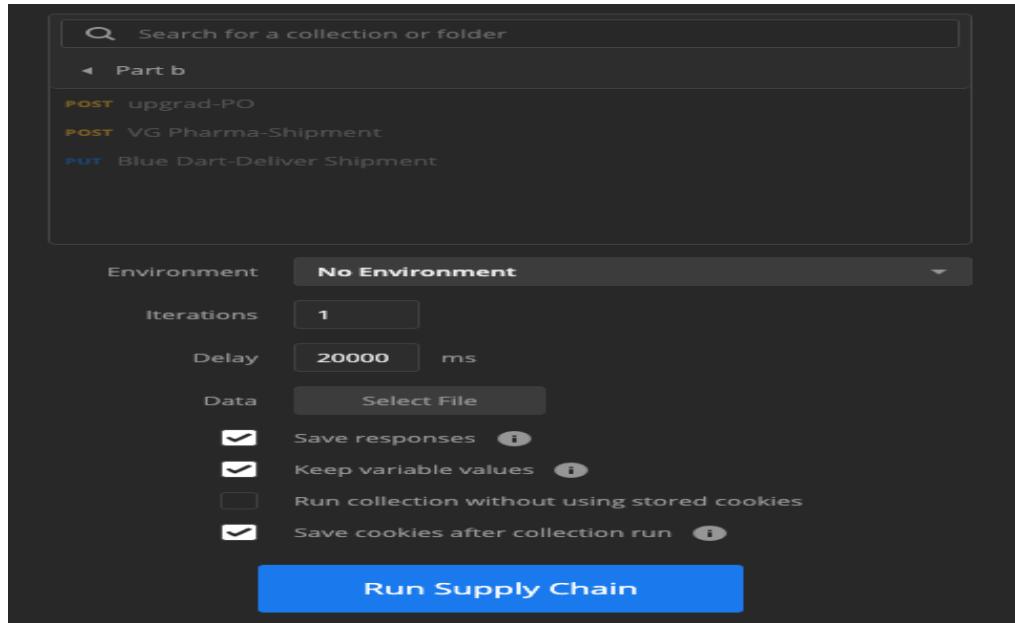
- Click on Collection Runner tab after Part a completes which will bring back collection runner window as shown below



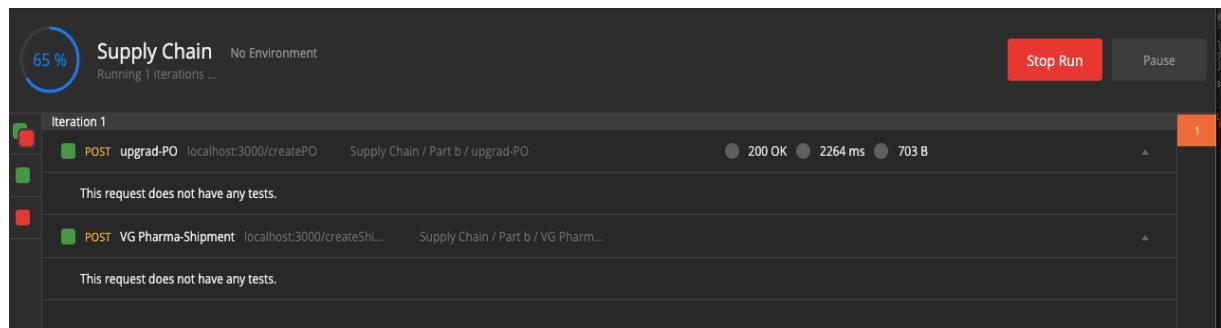
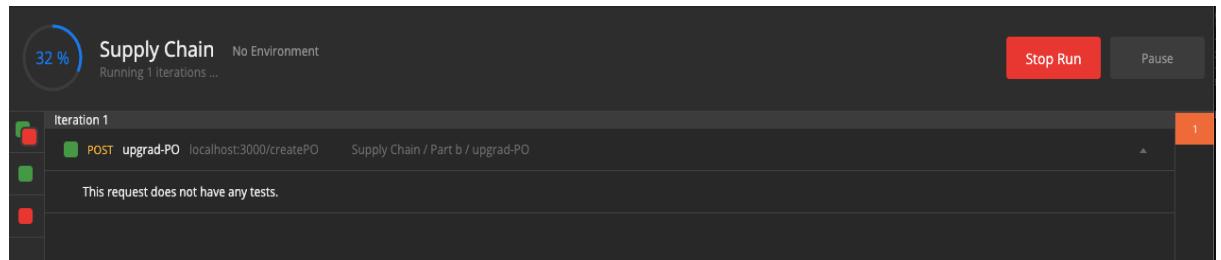
- Click on Part a folder to go back to Supply Chain collection as shown below
- Click on Part b folder under Supply Chain collection as shown below



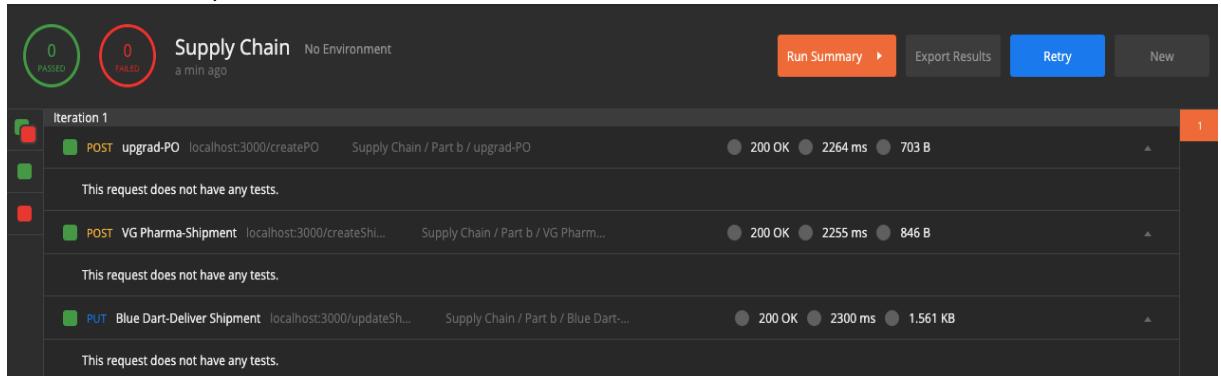
- Ensure settings are as shown below and note that delay between requests is set at 20000 ms which is 20 seconds



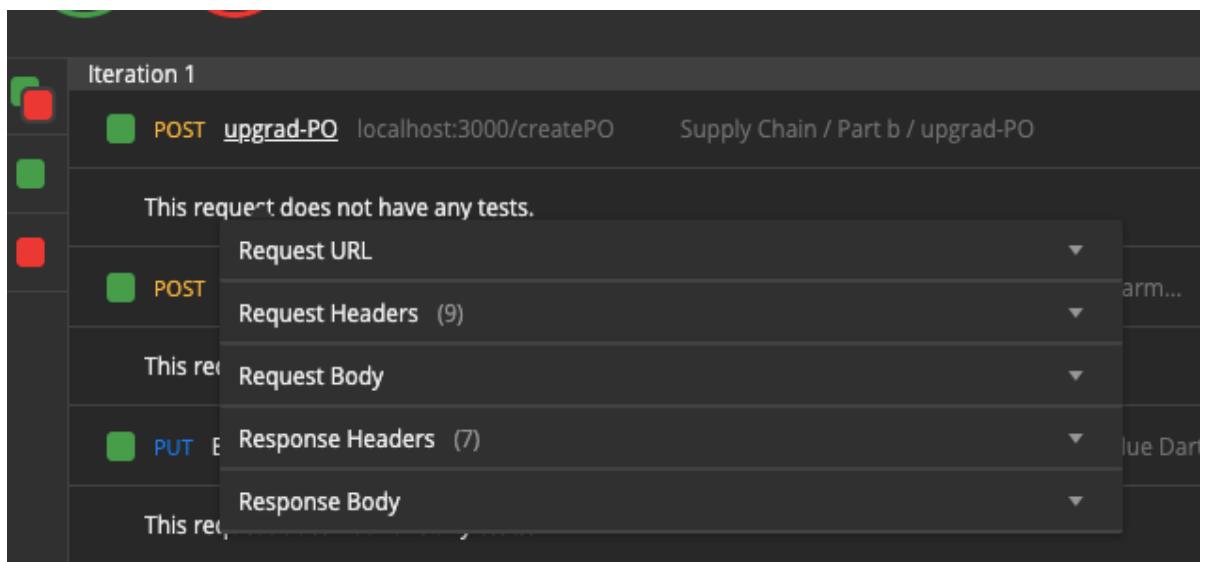
- Scroll down and click “Run Supply Chain” as shown below
- Click on Save response to store the response while running a collection
- This will open Run Results tab where run progress can be monitored



- Once the run completes, screen will look like below

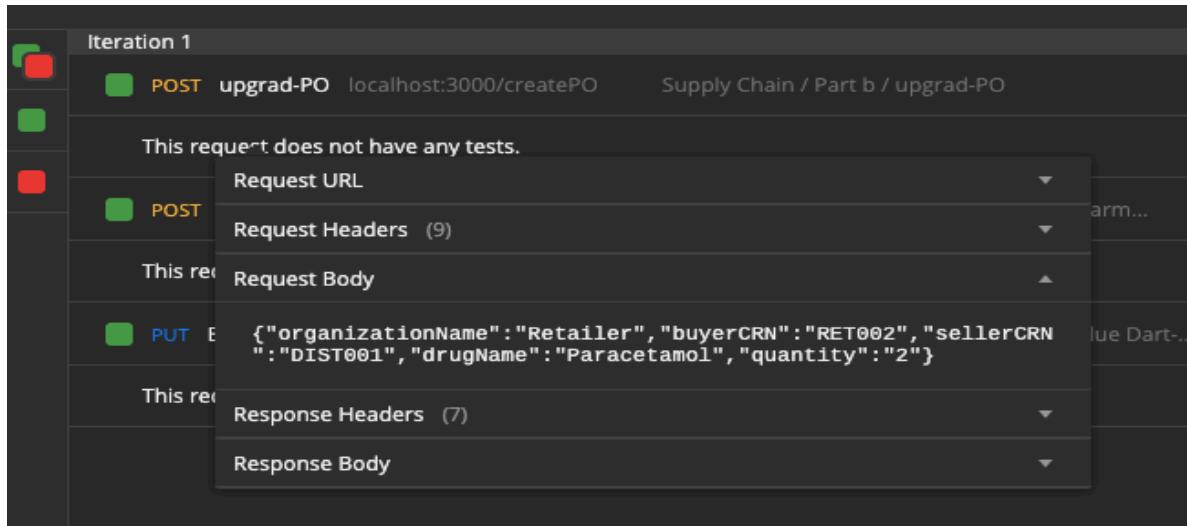


- The request parameters and response body for each request can be viewed by clicking on method name in Run Results tab as shown below



8.3.2.1. UPGRAD-PO

REQUEST:



Iteration 1

POST upgrad-PO localhost:3000/createPO Supply Chain / Part b / upgrad-PO

This request does not have any tests.

Request URL

Request Headers (9)

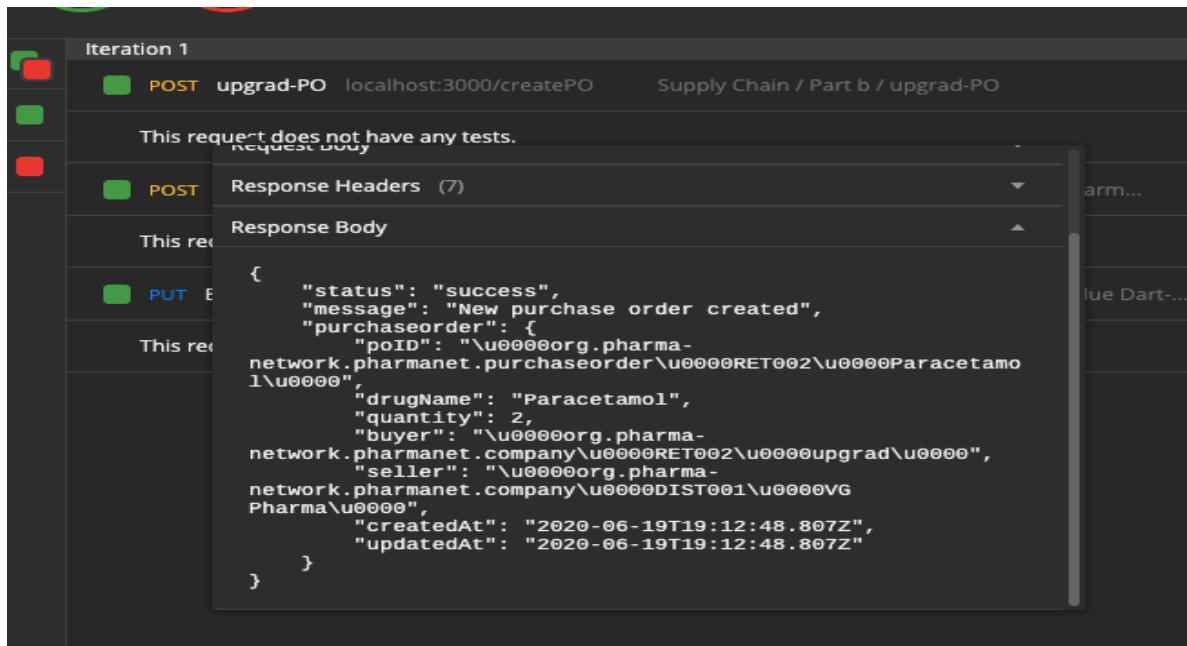
Request Body

```
{"organizationName": "Retailer", "buyerCRN": "RET002", "sellerCRN": "DIST001", "drugName": "Paracetamol", "quantity": "2"}
```

Response Headers (7)

Response Body

RESPONSE: Expected



Iteration 1

POST upgrad-PO localhost:3000/createPO Supply Chain / Part b / upgrad-PO

This request does not have any tests.

Response Headers (7)

Response Body

```
{
  "status": "success",
  "message": "New purchase order created",
  "purchaseorder": {
    "poID": "\u0000org.pharma-network.pharmanet.purchaseorder\u0000RET002\u0000Paracetamo1\u0000",
    "drugName": "Paracetamol",
    "quantity": 2,
    "buyer": "\u0000org.pharma-network.pharmanet.company\u0000RET002\u0000upgrad\u0000",
    "seller": "\u0000org.pharma-network.pharmanet.company\u0000DIST001\u0000VG Pharma\u0000",
    "createdAt": "2020-06-19T19:12:48.807Z",
    "updatedAt": "2020-06-19T19:12:48.807Z"
  }
}
```

8.3.2.2. VG PHARMA-SHIPMENT

REQUEST:

Iteration 1

POST upgrad-PO localhost:3000/createPO Supply Chain / Part b / upgrad-PO

This request does not have any tests.

POST VG Pharma-Shipment localhost:3000/createShi... Supply Chain / Part b / VG Pharm...

This request does not have any tests.

Request URL

PUT E

Request Headers (9)

This req Request Body

```
{"organizationName": "Distributor", "buyerCRN": "RET002", "drugNa  
me": "Paracetamol", "listOfAssets": "\\"  
["\\\"001\\\"", "\\\"002\\\""]\\\"", "transporterCRN": "TRA002"}
```

Response Headers (7)

Response Body

RESPONSE: Expected

Iteration 1

POST upgrad-PO localhost:3000/createPO Supply Chain / Part b / upgrad-PO

This request does not have any tests.

POST VG Pharma-Shipment localhost:3000/createShi... Supply Chain / Part b / VG Pharm...

This request does not have any tests.

Response Body

PUT E

```
{  
    "status": "success",  
    "message": "New shipment created",  
    "shipment": {  
        "shipmentID": "\u0000org.pharma-  
network.pharmanet.shipment\u0000RET002\u0000Paracetamol\u00  
00",  
        "creator": "\u0000org.pharma-  
network.pharmanet.company\u0000DIST001\u0000VG  
Pharma\u0000",  
        "assets": ["\u0000org.pharma-  
network.pharmanet.drug\u0000Paracetamol\u0000001\u0000",  
"\u0000org.pharma-  
network.pharmanet.drug\u0000Paracetamol\u0000002\u0000"],  
        "transporter": "\u0000org.pharma-  
network.pharmanet.company\u0000TRA002\u0000Blue  
Dart\u0000",  
        "status": "in-transit",  
        "createdAt": "2020-06-19T19:13:11.145Z",  
        "updatedAt": "2020-06-19T19:13:11.145Z"  
    }  
}
```

8.3.1.4. BLUE DART-DELIVERSHIPMENT

REQUEST:

Iteration 1

POST upgrad-PO localhost:3000/createPO Supply Chain / Part b / upgrad-PO
This request does not have any tests.

POST VG Pharma-Shipment localhost:3000/createShi... Supply Chain / Part b / VG Pharm...
This request does not have any tests.

PUT Blue Dart-Deliver Shipment localhost:3000/updateSh... Supply Chain / Part b / Blue Dart-...
This request does not have any tests.

Request URL

Request Headers (9)

Request Body

```
{"organizationName": "Transporter", "buyerCRN": "RET002", "drugName": "Paracetamol", "transporterCRN": "TRA002"}
```

Response Headers (7)

Response Body

RESPONSE: Expected

Iteration 1

POST upgrad-PO localhost:3000/createPO Supply Chain / Part b / upgrad-PO
This request does not have any tests.

POST VG Pharma-Shipment localhost:3000/createShi... Supply Chain / Part b / VG Pharm...
This request does not have any tests.

PUT Blue Dart-Deliver Shipment localhost:3000/updateSh... Supply Chain / Part b / Blue Dart-...
This request does not have any tests.

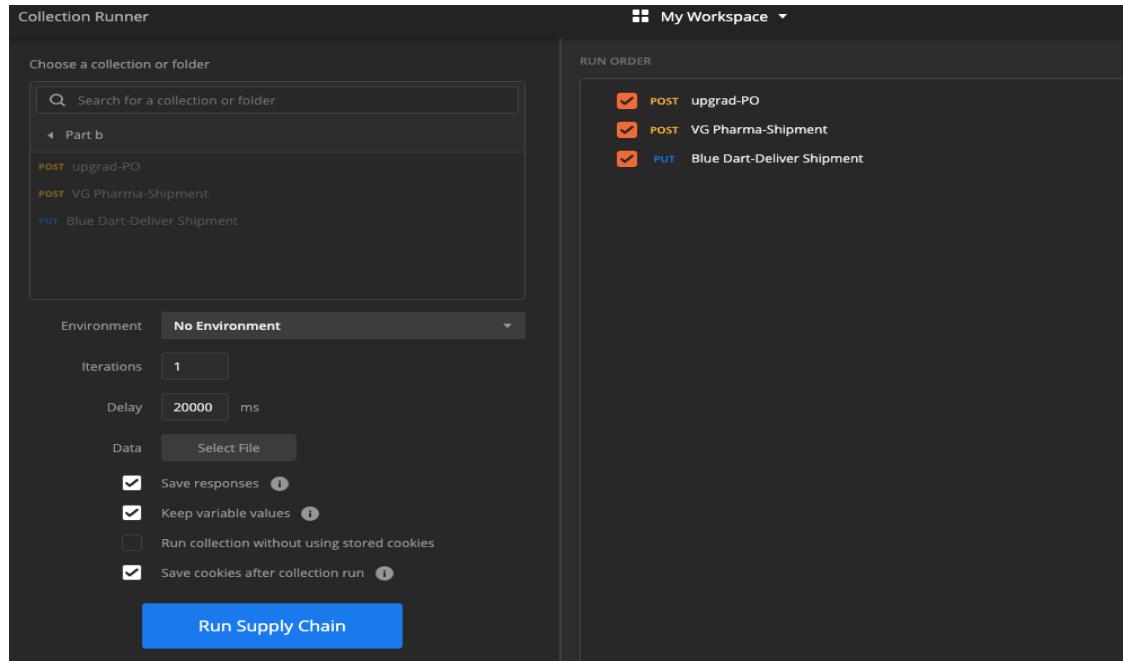
Response Body

```
{
  "status": "success",
  "message": "Shipment updated",
  "assets": [
    {
      "productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u0000001\u0000",
      "name": "Paracetamol",
      "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000sunPharma\u0000",
      "manufacturingDate": "2020-06-19T00:00:00.000Z",
      "expiryDate": "2022-09-18T00:00:00.000Z",
      "owner": "\u0000org.pharma-network.pharmanet.company\u0000RET002\u0000upgrad\u0000",
      "shipment": ["\u0000org.pharma-network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u0000", "\u0000org.pharma-network.pharmanet.shipment\u0000RET002\u0000Paracetamol\u0000"],
      "createdAt": "2020-06-19T18:55:45.970Z",
      "updatedAt": "2020-06-19T19:13:33.448Z"
    }
  ]
}
```

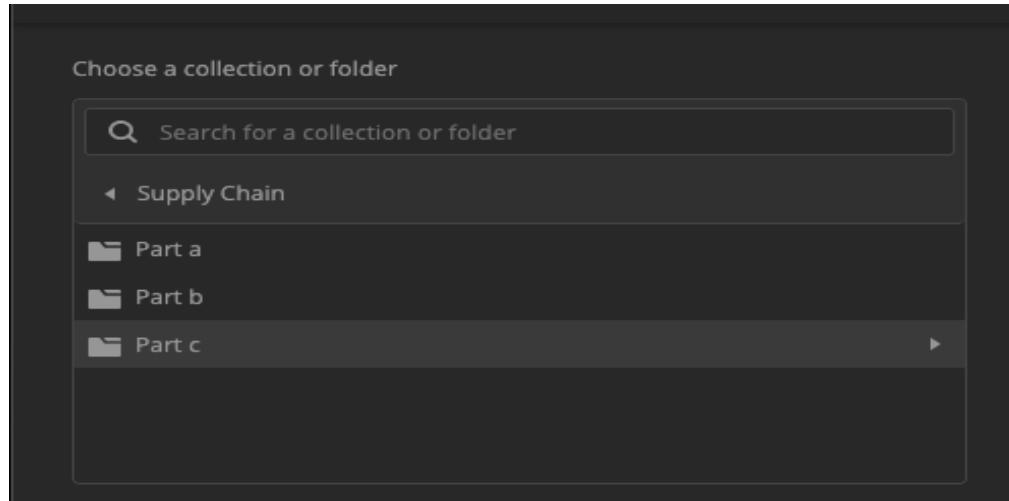
Iteration 1		
█	POST upgrad-PO localhost:3000/createPO	Supply Chain / Part b / upgrad-PO
█	This request does not have any tests.	
█	POST VG Pharma-Shipment localhost:3000/createShi...	Supply Chain / Part b / VG Pharm...
█	This request does not have any tests.	
█	PUT Blue Dart-Deliver Shipment localhost:3000/updateSh...	Supply Chain / Part b / Blue Dart-...
	<p>This request does not have any tests.</p> <pre> "id": "\u00000000-0000-0000-0000-000000000000", "network.pharmanet.shipment\u00000RET002\u0000Paracetamol\u0000", "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T19:13:33.448Z", }, { "productID": "\u00000org.pharma- network.pharmanet.drug\u0000Paracetamol\u0000002\u0000", "name": "Paracetamol", "manufacturer": "\u00000org.pharma- network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u00000org.pharma- network.pharmanet.company\u0000RET002\u0000upgrad\u0000", "shipment": ["\u00000org.pharma- network.pharmanet.shipment\u00000DIST001\u0000Paracetamol\u0000", "\u00000org.pharma- network.pharmanet.shipment\u00000RET002\u0000Paracetamol\u0000"], "createdAt": "2020-06-19T18:55:50.582Z", "updatedAt": "2020-06-19T19:13:33.451Z" } } </pre>	

8.3.3. PART C

- Click on Collection Runner tab after Part b completes which will bring back collection runner window as shown below



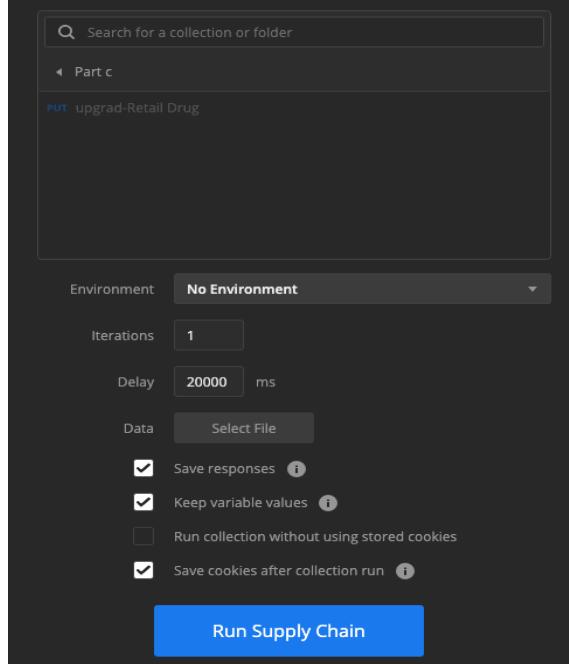
- Click on Part b folder to go back to Supply Chain collection as shown below
- Click on Part c folder under Supply Chain collection as shown below



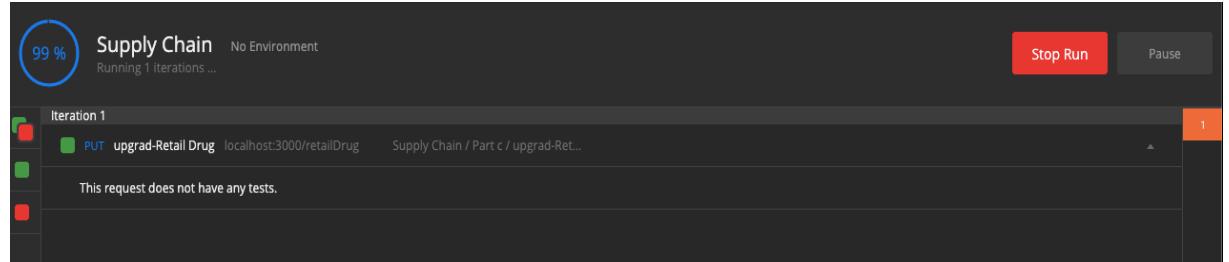
- Ensure settings are as shown below and note that there is only one test case in this folder

4

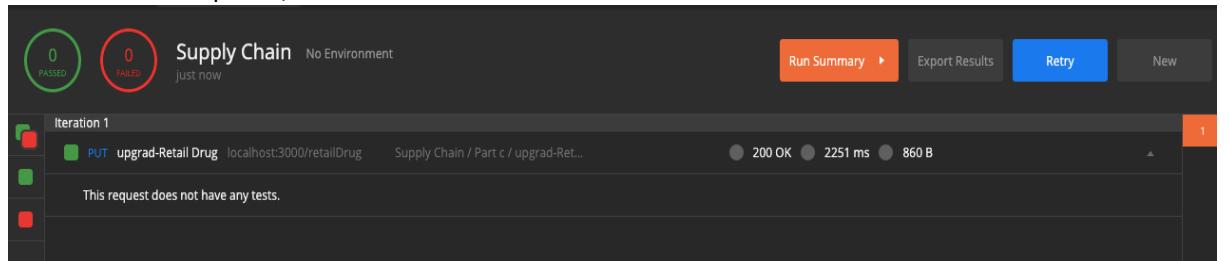
- Scroll down and click “Run Supply Chain” as shown below



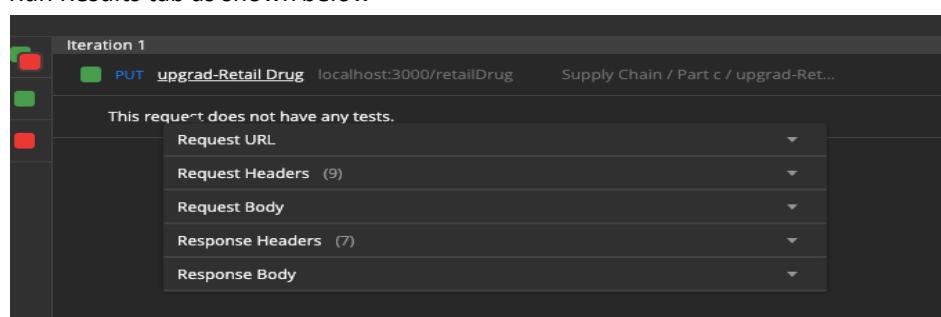
- This will open Run Results tab where run progress can be monitored



- Once the run completes, screen will look like below



- The request parameters and response body can be viewed by clicking on method name in Run Results tab as shown below



8.3.3.1. UPGRAD-RETAIL DRUG

REQUEST:

The screenshot shows the Postman interface with the following details:

- Iteration 1**
- Request URL:** `localhost:3000/retailDrug`
- Request Headers (9):** (List of headers is not visible)
- Request Body:**

```
{"organizationName": "Retailer", "drugName": "Paracetamol", "serialNo": "001", "retailerCRN": "RET002", "customerAadhar": "AAD001"}
```
- Response Headers (7):** (List of headers is not visible)
- Response Body:** (List of responses is not visible)

RESPONSE: Expected

The screenshot shows the Postman interface with the following details:

- Iteration 1**
- Request URL:** `localhost:3000/retailDrug`
- Response Body:**

```
{
  "status": "success",
  "message": "Drug retailed",
  "drug": {
    "productID": "\u0000org.pharma-network.pharanet.drug\u0000Paracetamol\u0000001\u0000",
    "name": "Paracetamol",
    "manufacturer": "\u0000org.pharma-network.pharanet.company\u0000MAN001\u0000Sun Pharma\u0000",
    "manufacturingDate": "2020-06-19T00:00:00.000Z",
    "expiryDate": "2022-09-18T00:00:00.000Z",
    "owner": "AAD001",
    "shipment": ["\u0000org.pharma-network.pharanet.shipment\u0000DIST001\u0000Paracetamol\u0000", "\u0000org.pharma-network.pharanet.shipment\u0000RET002\u0000Paracetamol\u0000"],
    "createdAt": "2020-06-19T18:55:45.970Z",
    "updatedAt": "2020-06-19T19:26:03.218Z"
  }
}
```

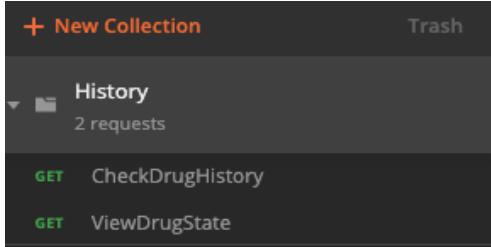
8.4. HISTORY TEST CASES

- This section explains various test cases available as part of "History" collection
- Although all test cases in a collection can be executed with the click of a single button,

DO NOT EXECUTE HISTORY TEST CASES IN A SINGLE GO

- The test cases part of this collection will be triggered separately since the requests are submitted to fabric as Consumer organization
- Since Consumer requests are not submitted so far, peer0 which is endorser of Consumer org will build its own chaincode container image where all dependencies of code are first installed through “npm install” before request is processed since pharma-network is running in Production mode**
- Hence history test cases are run one-by-one instead of all at once

- Close Postman collection runner opened in last section and come back to Postman tool
- Expand “History” collection by clicking on the small arrow before folder name as shown below



8.4.1. VIEWDRUGSTATE

- Click on ViewDrugState request to open it in a new tab and click “Send” to submit the request as shown below

The screenshot shows the Postman request details screen for the 'ViewDrugState' request. The method is set to 'GET', and the URL is 'localhost:3000/viewDrugCurrentState'. The 'Send' button is highlighted with a green box.

- Notice on terminal console output that the node application detects there is no wallet for “Consumer” organization and automatically creates it

```
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/consumer
.....Connecting to Fabric Gateway
.....Connecting to channel - pharmanetchannel
.....Connecting to Smart Contract in pharmanet chaincode
.....View Drug
.....Processing View Drug Transaction Response

{
  productID: '\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u0000001\u0000',
  name: 'Paracetamol',
  manufacturer: '\u0000org.pharma-network.pharmanet.company\u0000MAN001\u0000Sun Pharma\u0000',
  manufacturingDate: '2020-06-19T00:00:00.000Z',
  expiryDate: '2022-09-18T00:00:00.000Z',
  owner: 'AAD001',
  shipment: [
    '\u0000org.pharma-network.pharmanet.shipment\u0000DIST001\u0000Paracetamol\u0000',
    '\u0000org.pharma-network.pharmanet.shipment\u0000RET002\u0000Paracetamol\u0000'
  ],
  createdAt: '2020-06-19T18:55:45.970Z',
  updatedAt: '2020-06-19T19:26:03.218Z'
}

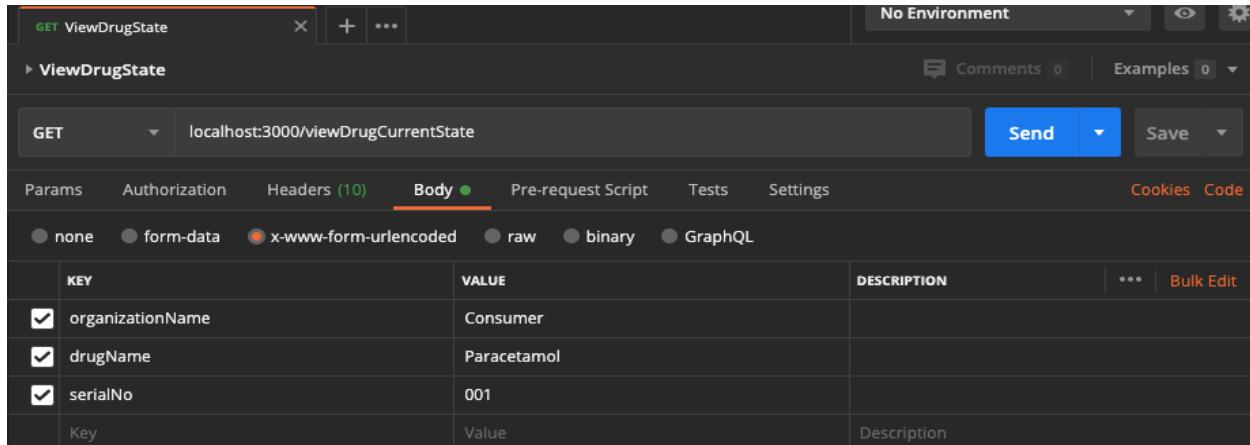
.....View Drug Transaction Complete!
.....Disconnecting from Fabric Gateway
Drug fetched
```

- Since this is the first request submitted as a Consumer, it is sent to peer0 of consumer organization for endorsement. peer0 of consumer then builds its own chaincode container image where all dependencies of code are first installed through “npm install” before request is processed since pharma-network is running in Production mode. Hence this request can take between 5 to 10 minutes to complete depending on machine configurations and network speed.

- After successful request execution, output is available on Postman under results section

Request details and expected output are explained below

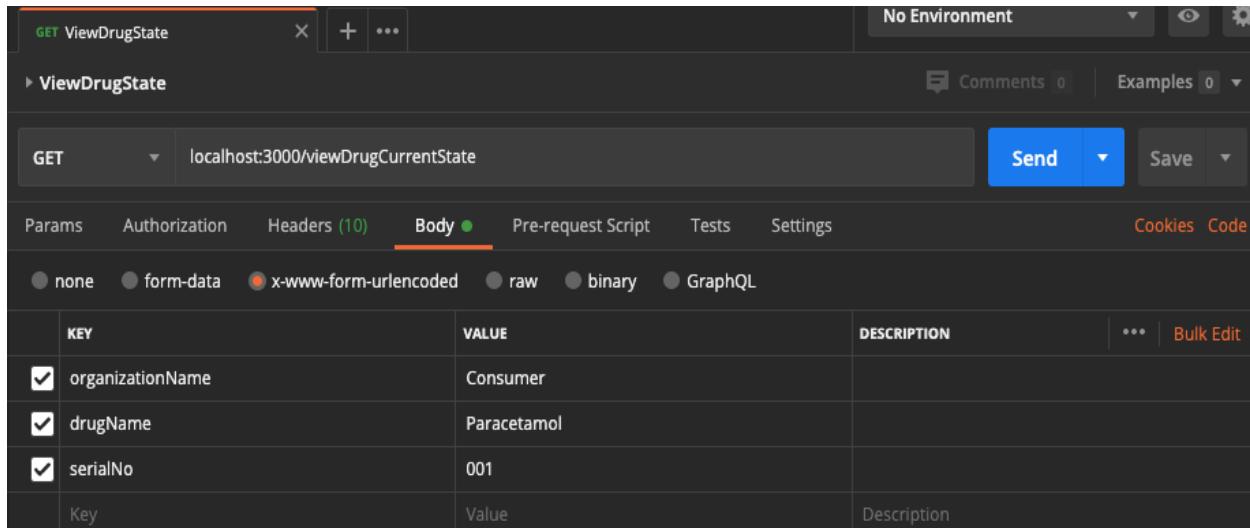
REQUEST:



The screenshot shows a Postman request configuration for a 'ViewDrugState' collection. The request is a GET to 'localhost:3000/viewDrugCurrentState'. The 'Body' tab is selected, showing an x-www-form-urlencoded body with three fields: organizationName (Consumer), drugName (Paracetamol), and serialNo (001). There is also a 'Key' row with 'Value' and 'Description' columns.

KEY	VALUE	DESCRIPTION
organizationName	Consumer	
drugName	Paracetamol	
serialNo	001	
Key	Value	Description

RESPONSE: Expected



The screenshot shows a Postman request configuration for a 'ViewDrugState' collection. The request is a GET to 'localhost:3000/viewDrugCurrentState'. The 'Body' tab is selected, showing an x-www-form-urlencoded body with three fields: organizationName (Consumer), drugName (Paracetamol), and serialNo (001). There is also a 'Key' row with 'Value' and 'Description' columns.

KEY	VALUE	DESCRIPTION
organizationName	Consumer	
drugName	Paracetamol	
serialNo	001	
Key	Value	Description

RESPONSE: Expected

Body Cookies Headers (7) Test Results Status: 200 OK Time: 2.37 s Size: 859 B Save Response ▾

Pretty Raw Preview Visualize JSON

```

1 [
2   "status": "success",
3   "message": "Drug fetched",
4   "drug": {
5     "productID": "@org.pharma-network.pharmanet.drug@Paracetamol@0001@",
6     "name": "Paracetamol",
7     "manufacturer": "@org.pharma-network.pharmanet.company@MAN001@Sun Pharma@",
8     "manufacturingDate": "2020-06-19T00:00:00.000Z",
9     "expiryDate": "2022-09-18T00:00:00.000Z",
10    "owner": "AAD001",
11    "shipment": ["@org.pharma-network.pharmanet.shipment@DIST001@Paracetamol@", "@org.pharma-network.pharmanet.shipment@RET002@Paracetamol@"]
12  ],
13  "createdAt": "2020-06-19T18:55:45.970Z",
14  "updatedAt": "2020-06-19T19:26:03.218Z"
15 }
16 ]

```

8.4.2. CHECKDRUGHISTORY

- Click on CheckDrugHistory request to open it in a new tab and click “Send” to submit the request as shown below

The screenshot shows the Postman application interface. A specific request named 'CheckDrugHistory' is selected. The 'Send' button, located at the top right of the request details panel, is highlighted with a green rectangular box. The URL 'localhost:3000/viewHistory' is visible in the 'Body' section. Other tabs like 'Params', 'Authorization', and 'Headers' are also visible.

- Notice on terminal console output that the node application detects wallet is already available for “Consumer” organization (created from previous ViewDrugState request) and does not create it again

```

    "does not exist!!!
....Either wallet does not exist or does not have expected number of files
....Creating wallet now for Consumer
....Creating wallet now for Consumer
/Users/manikandan/Desktop/Capstone_Project_Manikandan/pharma-net/Application/identity/consumer
....Connecting to Fabric Gateway
....Connecting to channel - pharmachannel
....Connecting to Smart Contract in pharmanet chaincode
....View Lifecycle
....View Lifecycle Transaction Response

[ {
    TransactionId: '85fc694080176d872cf2978ffbcc8cc116705e2761d8f3ecbcebf8e9f0494767',
    Timestamp: '2020-06-19T18:55:45.867Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "shipment": [], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T18:55:45.970Z" },
],
{
    TransactionId: 'a64bf10195c8cc40fb60828fe9ca8b195756653ead220cfce0084408ec',
    Timestamp: '2020-06-19T18:59:14.998Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharma-network.pharmanet.company\u0000TRA001\u00000FedEx\u0000", "shipment": [], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T18:59:14.998Z" },
],
{
    TransactionId: 'd2f94e06294b15b192e1f288f28730629904cf11774536b5be85073fbe5efabc',
    Timestamp: '2020-06-19T18:59:15.735Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharmanet.shipment\u0000DIST001\u00000Paracetamol\u0000", "shipment": [{"\u0000org.pharma-network.pharmanet.shipment\u0000DIST001\u00000Paracetamol\u0000"}], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T18:59:15.735Z" },
],
{
    TransactionId: 'e6e5057a7909185fa01b41bef1c50d2dc51cdcad383bcb1le81dd388a544d11',
    Timestamp: '2020-06-19T19:13:11.039Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharma-network.pharmanet.company\u0000TRA002\u00000Blue Dart\u0000", "shipment": [{"\u0000org.pharma-network.pharmanet.shipment\u0000DIST001\u00000Paracetamol\u0000"}], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T19:13:11.039Z" },
],
{
    TransactionId: 'b7528b0fb1eb1db3d9c6e1b77db6bde0e6c3b3297bcde86d048fb33d52c645cf4',
    Timestamp: '2020-06-19T19:13:33.339Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharma-network.pharmanet.company\u0000TRA001\u00000FedEx\u0000", "shipment": [{"\u0000org.pharma-network.pharmanet.shipment\u0000DIST001\u00000Paracetamol\u0000"}], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T19:13:33.339Z" },
],
{
    TransactionId: 'a414bf1e41925c8ce4885fb60820fe9ca8b195756653ead220cfce0084408ec',
    Timestamp: '2020-06-19T18:59:14.398Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharma-network.pharmanet.company\u0000TRA001\u00000FedEx\u0000", "shipment": [{"\u0000org.pharma-network.pharmanet.shipment\u0000DIST001\u00000Paracetamol\u0000"}], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T18:59:14.398Z" },
],
{
    TransactionId: 'd2f94e06294b15b192e1f288f28730629904cf11774536b5be85073fbe5efabc',
    Timestamp: '2020-06-19T18:59:15.735Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharma-network.pharmanet.company\u0000TRA002\u00000Blue Dart\u0000", "shipment": [{"\u0000org.pharma-network.pharmanet.shipment\u0000DIST001\u00000Paracetamol\u0000"}], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T18:59:15.735Z" },
],
{
    TransactionId: 'e6e5057a7909185fa01b41bef1c50d2dc51cdcad383bcb1le81dd388a544d11',
    Timestamp: '2020-06-19T19:13:11.039Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharma-network.pharmanet.company\u0000TRA002\u00000Blue Dart\u0000", "shipment": [{"\u0000org.pharma-network.pharmanet.shipment\u0000DIST001\u00000Paracetamol\u0000"}], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T19:13:11.039Z" },
],
{
    TransactionId: 'b7528b0fb1eb1db3d9c6e1b77db6bde0e6c3b3297bcde86d048fb33d52c645cf4',
    Timestamp: '2020-06-19T19:13:33.339Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "\u0000org.pharma-network.pharmanet.company\u0000TRA001\u00000Paracetamol\u0000", "shipment": [{"\u0000org.pharma-network.pharmanet.shipment\u0000DIST001\u00000Paracetamol\u0000"}], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T19:13:33.339Z" },
],
{
    TransactionId: '25cc32d3ae382848fe1f45eb6476d84bc90f3c113639fc2af5865b2776518f7d5',
    Timestamp: '2020-06-19T19:26:03.174Z',
    Data: {"productID": "\u0000org.pharma-network.pharmanet.drug\u0000Paracetamol\u00000001\u00000001", "name": "Paracetamol", "manufacturer": "\u0000org.pharma-network.pharmanet.company\u0000MAN001\u00000Sun Pharma\u0000", "manufacturingDate": "2020-06-19T00:00:00.000Z", "expiryDate": "2022-09-18T00:00:00.000Z", "owner": "AAD001", "shipment": [{"\u0000org.pharma-network.pharmanet.shipment\u0000RET002\u00000Paracetamol\u0000"}], "createdAt": "2020-06-19T18:55:45.970Z", "updatedAt": "2020-06-19T19:26:03.174Z" }
]

....View Lifecycle Transaction Complete!
....Disconnecting from Fabric Gateway
Drug history fetched

```

- Since this is the second request submitted as a Consumer which is sent to peer0 of Consumer organization for endorsement, its corresponding chaincode container is already created from previous ViewDrugState request and hence this execution completes in few seconds.
- After successful request execution, output is available on Postman under results section

Request details and expected output are explained below

REQUEST:

The screenshot shows the Postman application interface. At the top, there's a header bar with tabs for 'CheckDrugHistory' (selected), 'Comments 0', 'Examples 0', and environment dropdowns. Below the header, the main area shows a 'GET' request to 'localhost:3000/viewHistory'. The 'Body' tab is selected, showing a table with three rows of form-data parameters:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> organizationName	Consumer	
<input checked="" type="checkbox"/> drugName	Paracetamol	
<input checked="" type="checkbox"/> serialNo	001	
Key	Value	Description

At the bottom right of the body table, there are buttons for '... | Bulk Edit'.

RESPONSE: Expected

Body Cookies Headers (7) Test Results Status: 200 OK Time: 2 m 33.30 s Size: 4.53 KB Save Response ▾

Pretty Raw Preview Visualize JSON ⚡

```

4   "drugHistory": [
5     {
6       "TransactionId": "85fc604080176d872cf2978ffbcc8cc116705e2761d8f3ecbcebf8e9f0494767",
7       "Timestamp": "2020-06-19T18:55:45.867Z",
8       "Data": "{\"productID\":\"\\@org.pharma-network.pharmanet.drug\\@Paracetamol\\@001\\@\\",
9         \"name\":\"Paracetamol\\\", \"manufacturer\":\"\\@org.pharma-network.pharmanet.company\\@MAN001\\@Sun
10        Pharma\\@\\\", \"manufacturingDate\":\"2020-06-19T00:00:00.000Z\\",
11        \"expiryDate\":\"2022-09-18T00:00:00.000Z\\\", \"owner\":\"\\@org.pharma-network.pharmanet.
12        company\\@MAN001\\@Sun Pharma\\@\\\", \"shipment\":[], \"createdAt\":\"2020-06-19T18:55:45.970Z\\",
13        \"updatedAt\":\"2020-06-19T18:55:45.970Z\\\"}"
14     },
15   },
16   {
17     "TransactionId": "a414bf1e41925c8ce4885fb60820fe9ca8b1957566533ead220cfce0084408ec",
18     "Timestamp": "2020-06-19T18:59:14.398Z",
19     "Data": "{\"productID\":\"\\@org.pharma-network.pharmanet.drug\\@Paracetamol\\@001\\@\\",
20       \"name\":\"Paracetamol\\\", \"manufacturer\":\"\\@org.pharma-network.pharmanet.company\\@MAN001\\@Sun
21        Pharma\\@\\\", \"manufacturingDate\":\"2020-06-19T00:00:00.000Z\\",
22        \"expiryDate\":\"2022-09-18T00:00:00.000Z\\\", \"owner\":\"\\@org.pharma-network.pharmanet.
23        company\\@TRA001\\@FedEx\\@\\\", \"shipment\":[], \"createdAt\":\"2020-06-19T18:55:45.970Z\\",
24        \"updatedAt\":\"2020-06-19T18:59:14.495Z\\\"}"
25   },
26   {
27     "TransactionId": "d2f94e06294b15b192e1f280f28730629904cf11774536b5be85073fbe5efabc",
28     "Timestamp": "2020-06-19T18:59:36.735Z",
29     "Data": "{\"productID\":\"\\@org.pharma-network.pharmanet.drug\\@Paracetamol\\@001\\@\\",
30       \"name\":\"Paracetamol\\\", \"manufacturer\":\"\\@org.pharma-network.pharmanet.company\\@MAN001\\@Sun
31        Pharma\\@\\\", \"manufacturingDate\":\"2020-06-19T00:00:00.000Z\\",
32        \"expiryDate\":\"2022-09-18T00:00:00.000Z\\\", \"owner\":\"\\@org.pharma-network.pharmanet.
33        company\\@TRA002\\@Blue Dart\\@\\\", \"shipment\":[], \"createdAt\":\"2020-06-19T18:55:45.970Z\\",
34        \"updatedAt\":\"2020-06-19T19:13:11.138Z\\\"}"
35   }
36 ]

```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 2 m 33.30 s Size: 4.53 KB Save Response ▾

Pretty Raw Preview Visualize JSON ⚡

```

21   "TransactionId": "e6e5057a7909185fa01b41bef1c50d2dc51cdcad383bcb11e81dd388a544d11",
22   "Timestamp": "2020-06-19T19:13:11.039Z",
23   "Data": "{\"productID\":\"\\@org.pharma-network.pharmanet.drug\\@Paracetamol\\@001\\@\\",
24     \"name\":\"Paracetamol\\\", \"manufacturer\":\"\\@org.pharma-network.pharmanet.company\\@MAN001\\@Sun
25        Pharma\\@\\\", \"manufacturingDate\":\"2020-06-19T00:00:00.000Z\\",
26        \"expiryDate\":\"2022-09-18T00:00:00.000Z\\\", \"owner\":\"\\@org.pharma-network.pharmanet.
27        company\\@TRA002\\@Blue Dart\\@\\\", \"shipment\":[], \"createdAt\":\"2020-06-19T18:55:45.970Z\\",
28        \"updatedAt\":\"2020-06-19T19:13:11.138Z\\\"}"
29   },
30   {
31     "TransactionId": "b7528b0f1be1db3d9c6e1b77db6bde0e6c3b3297bcde86d048fb33d52c645cf4",
32     "Timestamp": "2020-06-19T19:13:33.339Z",
33     "Data": "{\"productID\":\"\\@org.pharma-network.pharmanet.drug\\@Paracetamol\\@001\\@\\",
34       \"name\":\"Paracetamol\\\", \"manufacturer\":\"\\@org.pharma-network.pharmanet.company\\@MAN001\\@Sun
35        Pharma\\@\\\", \"manufacturingDate\":\"2020-06-19T00:00:00.000Z\\",
36        \"expiryDate\":\"2022-09-18T00:00:00.000Z\\\", \"owner\":\"\\@org.pharma-network.pharmanet.
37        company\\@RET002\\@upgrade\\@\\\", \"shipment\":[], \"createdAt\":\"2020-06-19T18:55:45.970Z\\",
38        \"updatedAt\":\"2020-06-19T19:13:33.448Z\\\"}"
39   },
40   {
41     "TransactionId": "25cc32d3ae538248fe1f45eb6476d84bc90f3c113639fc2af585b2776518ffd5",
42     "Timestamp": "2020-06-19T19:26:03.174Z",
43     "Data": "{\"productID\":\"\\@org.pharma-network.pharmanet.drug\\@Paracetamol\\@001\\@\\",
44       \"name\":\"Paracetamol\\\", \"manufacturer\":\"\\@org.pharma-network.pharmanet.company\\@MAN001\\@Sun
45        Pharma\\@\\\", \"manufacturingDate\":\"2020-06-19T00:00:00.000Z\\",
46        \"expiryDate\":\"2022-09-18T00:00:00.000Z\\\", \"owner\":\"\\@org.pharma-network.pharmanet.
47        company\\@RET002\\@Paracetamol\\@\\\", \"shipment\":[], \"createdAt\":\"2020-06-19T18:55:45.970Z\\",
48        \"updatedAt\":\"2020-06-19T19:26:03.174Z\\\"}"
49   }
50 ]

```

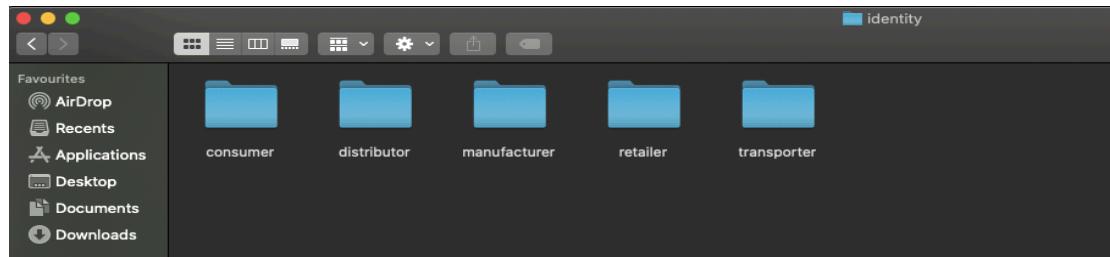
```

GET      localhost:3000/viewHistory      Send      Save
Pretty  Raw   Preview  Visualize  JSON  ⚙️
24      },    {
25      {
26          "TransactionId": "b7528b0f1be1db3d9c6e1b77db6bde0e6c3b3297bcde86d048fb33d52c645cf4",
27          "Timestamp": "2020-06-19T19:13:33.39Z",
28          "Data": "{\"productID\":\"@org.pharma-network.pharmanet.Paracetamol@001@\","
29          \"name\":\"Paracetamol\", \"manufacturer\":\"@org.pharma-network.pharmanet.company@MAN001@Sun Pharma@\", \"manufacturingDate\":\"2020-06-19T00:00:00.000Z\","
30          \"expiryDate\":\"2022-09-18T00:00:00.000Z\", \"owner\":\"@org.pharma-network.pharmanet."
31          \"shipment@DIST002@Upgrad@\", \"shipment@\": [\"@org.pharma-network.pharmanet."
32          \"shipment@DIST001@Paracetamol@\", \"@org.pharma-network.pharmanet."
33          \"shipment@RET002@Paracetamol@\"], \"createdAt\":\"2020-06-19T18:55:45.970Z\","
34          \"updatedAt\":\"2020-06-19T19:13:33.448Z\"}"
35      },
36      ]

```

8.5. VIEW WALLETS

- Navigate to “identity” folder inside the “application” folder of project in file explorer to view the wallets generate by node application as shown below



- Sample structure of one the wallet folders is shown below

