

Pharma-Network

Created By: Soumik Saha

1. Introduction

Pharma network is a platform to tackle drug counterfeiting. The Network will consist of following participant :

- Manufacturers
- Distributors
- Retailers
- Consumers
- Transporters

This document will guide you through topics, which will help you to understand various functionalities available in this network.

- Prerequisite
- Installing Libraries
- Starting the Hyperledger fabric based Pharma Network
- Starting the Client Side REST API application
- Running the Test Cases
- Clearing the setup files to re-launch the network

2. Prerequisite

The application was developed using the following version of node, npm, Hyperledger fabric and docker

- Node version : v18.15.0
- NPM version : 9.5.0
- HyperLedger Fabric : 2.4
- Docker Version : 24.0.4 build 3713ee1
- Docker Compose version : v2.15.1

```
soumik@SoumikPC:~$ node --version
v18.15.0
soumik@SoumikPC:~$ npm --version
9.5.0
soumik@SoumikPC:~$ docker version
Client: Docker Engine - Community
  Cloud integration: v1.0.31
  Version:          24.0.4
  API version:     1.41 (downgraded from 1.43)
  Go version:      go1.20.5
  Git commit:      3713ee1
  Built:           Fri Jul  7 14:50:55 2023
  OS/Arch:         linux/amd64
  Context:         desktop-linux

Server: Docker Desktop 4.17.0 (99724)
Engine:
  Version:          20.10.23
  API version:     1.41 (minimum version 1.12)
  Go version:      go1.18.10
  Git commit:      6051f14
  Built:           Thu Jan 19 17:32:04 2023
  OS/Arch:         linux/amd64
  Experimental:   false
containerd:
  Version:          1.6.18
  GitCommit:        2456e983eb9e37e47538f59ea18f2043c9a73640
runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
soumik@SoumikPC:~$ docker --version
Docker version 24.0.4, build 3713ee1
soumik@SoumikPC:~$ docker-compose --version
Docker Compose version v2.15.1
soumik@SoumikPC:~$ 
```

Please ensure that the prerequisite are met before going ahead with the other setup process.

3. Installing Libraries

Please Note : Since, the project folder can be different based on where you have placed the project, henceforth the base path till the project folder will be denoted as {Path_to_Project} (this does not include the project folder name).

- Installing the Libraties for Client application :

Once the above modules are installed, open terminal and navigate to the project folder {Path_to_Project}/pharma-network/application.

Run the command : npm install

This will install all the required libraries for the client side REST API application.

Please refer to the below image:

```
● soumik@SoumikPC:~/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/application$ npm install
Debugger listening on ws://127.0.0.1:41053/91a72cd6-9a34-465b-9338-eae1f5b0b663
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.

added 262 packages, and audited 263 packages in 11s
39 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
Waiting for the debugger to disconnect...
```

- Installing the libraries for Chaincodes:

To install all the libraries for chaincode, in the same terminal/new terminal, navigate to the path {Path_to_Project}/pharma-network/chaincodes.

Run the command : npm install

This will install all the required libraries for chaincodes.

```
● soumik@SoumikPC:~/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/chaincodes$ npm install
Debugger listening on ws://127.0.0.1:44607/580d901f-9f53-4946-a046-fc2c6d4e2664
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.

added 365 packages, and audited 366 packages in 7s
47 packages are looking for funding
  run `npm fund` for details

3 high severity vulnerabilities

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
Waiting for the debugger to disconnect...
```

If you have reached till here, you have successfully, installed all the required libraires to run the network.

4. Starting the Hyperledger fabric based Pharma Network

To start the hyperledger fabric network, navigate to the folder , {Path_to_Project}/pharma-network/test-network-man

Inside this folder, you will find the following folders :-

- **AnchorPeerScripts** : This folder contains the script to create anchor peers.
- **Configtx** : Contains the configuration details for fabric network
- **Dockers**: Contains the compose files to create the required dockers
- **Organizations**: Contains the crypto libraries to create the required organizations

Also the folder contains multiple .sh scripts which are used to start the network.

Note, before you start to execute the shell script, please make sure that the script are executable. The simple way to do this is :-

- Open File Explorer
- Navigate to {Path_to_Project}/pharma-network/test-network-man
- Right click on the .sh files and select properties
- Select the permission tab
- On the bottom select the checkbox, Allow executing file as program.

Note, the above set of actions need to be done for the following set of files :-

- testFile.sh
- startNetwork.sh
- deployChaincode.sh
- create_join_Channel.sh
- clearSetup.sh

Once the above steps are completed,we can start the network.

To start the network, please make sure that in the terminal, you are in the folder {Path_to_Project}/pharma-network/test-network-man, and run the script **./startNetwork.sh**.

This script will bring up our entire network, which includes :-

- Creating all the required crypto materials
- Starting the all CA,Orderer,Peers and Chaincode dockers
- Creating channels
- Joining the peer in the channel
- Creating anchor peers
- Initializing Chaincode
- Approving Chaincode
- Deploying Chaincode

Please find the below screenshot :-

```
* soumik@SoumikPC-1:~/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man$ ./startNetwork.sh
=====
Setting Path =====
=====
Creating Folders =====
=====
Generating crypto Materials =====
Consumer.pharma-network.com
Distributor.pharma-network.com
Manufacturer.pharma-network.com
Retailer.pharma-network.com
Transporter.pharma-network.com
=====
Generating Genesis Block for Orderer =====
2023-08-09 15:31:10.011 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.015 15T 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: solo
2023-08-09 15:31:10.015 15T 0003 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
2023-08-09 15:31:10.016 15T 0004 INFO [common.tools.configtxgen] doOutputBlock -> Generating genesis block
2023-08-09 15:31:10.016 15T 0005 INFO [common.tools.configtxgen] doOutputBlock -> Creating system channel genesis block
2023-08-09 15:31:10.017 15T 0006 INFO [common.tools.configtxgen] doOutputBlock -> Writing genesis block
=====
Generating JSON Output for Genesis Block =====
2023-08-09 15:31:10.033 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.033 15T 0002 INFO [common.tools.configtxgen] doInspectBlock -> Inspecting block
2023-08-09 15:31:10.033 15T 0003 INFO [common.tools.configtxgen] doInspectBlock -> Parsing genesis block
=====
Generating Peer =====
2023-08-09 15:31:10.036 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.036 15T 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
2023-08-09 15:31:10.066 15T 0003 INFO [common.tools.configtxgen] doOutputChannelCreateTx -> Generating new channel configtx
2023-08-09 15:31:10.068 15T 0004 INFO [common.tools.configtxgen] doOutputChannelCreateTx -> Writing new channel tx
=====
Generating JSON Output for the Pharma Channel =====
2023-08-09 15:31:10.084 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.084 15T 0002 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Inspecting transaction
2023-08-09 15:31:10.084 15T 0003 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Parsing transaction
=====
Updating Anchor Peers =====
2023-08-09 15:31:10.103 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.103 15T 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
2023-08-09 15:31:10.107 15T 0003 INFO [common.tools.configtxgen.localconfig] doOutputAnchorPeerUpdate -> Generating anchor peer update
2023-08-09 15:31:10.107 15T 0004 INFO [common.tools.configtxgen] doOutputAnchorPeerUpdate -> Writing anchor peer update
2023-08-09 15:31:10.109 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.120 15T 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
2023-08-09 15:31:10.120 15T 0003 INFO [common.tools.configtxgen] doOutputAnchorPeerUpdate -> Generating anchor peer update
2023-08-09 15:31:10.120 15T 0004 INFO [common.tools.configtxgen.localconfig] doOutputAnchorPeerUpdate -> Writing anchor peer update
2023-08-09 15:31:10.131 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.131 15T 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
2023-08-09 15:31:10.151 15T 0003 INFO [common.tools.configtxgen.localconfig] doOutputAnchorPeerUpdate -> Generating anchor peer update
2023-08-09 15:31:10.151 15T 0004 INFO [common.tools.configtxgen] doOutputAnchorPeerUpdate -> Writing anchor peer update
2023-08-09 15:31:10.154 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.154 15T 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
2023-08-09 15:31:10.154 15T 0003 INFO [common.tools.configtxgen] doOutputAnchorPeerUpdate -> Generating anchor peer update
2023-08-09 15:31:10.154 15T 0004 INFO [common.tools.configtxgen.localconfig] doOutputAnchorPeerUpdate -> Writing anchor peer update
2023-08-09 15:31:10.160 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.173 15T 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
2023-08-09 15:31:10.173 15T 0003 INFO [common.tools.configtxgen] doOutputAnchorPeerUpdate -> Generating anchor peer update
2023-08-09 15:31:10.173 15T 0004 INFO [common.tools.configtxgen.localconfig] doOutputAnchorPeerUpdate -> Writing anchor peer update
2023-08-09 15:31:10.189 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.193 15T 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
2023-08-09 15:31:10.193 15T 0003 INFO [common.tools.configtxgen] doOutputAnchorPeerUpdate -> Generating anchor peer update
2023-08-09 15:31:10.193 15T 0004 INFO [common.tools.configtxgen.localconfig] doOutputAnchorPeerUpdate -> Writing anchor peer update
=====
Generating JSON =====
2023-08-09 15:31:10.210 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.210 15T 0002 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Inspecting transaction
2023-08-09 15:31:10.210 15T 0003 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Parsing transaction
2023-08-09 15:31:10.220 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.220 15T 0002 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Inspecting transaction
2023-08-09 15:31:10.220 15T 0003 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Parsing transaction
2023-08-09 15:31:10.243 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.244 15T 0002 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Inspecting transaction
2023-08-09 15:31:10.244 15T 0003 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Parsing transaction
2023-08-09 15:31:10.263 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.263 15T 0002 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Inspecting transaction
2023-08-09 15:31:10.263 15T 0003 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Parsing transaction
2023-08-09 15:31:10.279 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.279 15T 0002 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Inspecting transaction
2023-08-09 15:31:10.279 15T 0003 INFO [common.tools.configtxgen] doInspectChannelCreateTx -> Parsing transaction
=====
Generating Information Organizations =====
2023-08-09 15:31:10.293 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.293 15T 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: solo
2023-08-09 15:31:10.293 15T 0003 INFO [common.tools.configtxgen.localconfig] LoadTopLevel -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
2023-08-09 15:31:10.316 15T 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-08-09 15:31:10.321 15T 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: solo
2023-08-09 15:31:10.321 15T 0003 INFO [common.tools.configtxgen.localconfig] LoadTopLevel -> Loaded configuration: /home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/test-network-man/configtx/configtx.yaml
```

```

# Volume "dockers.peer0.Manufacturer.pharma-network.com" Created
# Volume "dockers.peer0.Consumer.pharma-network.com" Created
# Volume "dockers.peer0.Distributor.pharma-network.com" Created
[+] Running 26/30 peer0.Distributor.pharma-network.com
# Network pharma-network
# Volume "dockers.orderer.pharma-network.com" Created
# Volume "dockers.peer0.Consumer.pharma-network.com" Created
# Volume "dockers.peer1.Retailer.pharma-network.com" Created
# Volume "dockers.peer1.Manufacturer.pharma-network.com" Created
# Volume "dockers.peer1.Transporter.pharma-network.com" Created
# Volume "dockers.peer1.Distributor.pharma-network.com" Created
[+] Running 28/30s peer0.Distributor.pharma-network.com
# Network pharma-network
# Volume "dockers.orderer.pharma-network.com" Created
# Volume "dockers.peer0.Consumer.pharma-network.com" Created
# Volume "dockers.peer1.Retailer.pharma-network.com" Created
# Volume "dockers.peer0.Manufacturer.pharma-network.com" Created
# Volume "dockers.peer1.Consumer.pharma-network.com" Created
# Volume "dockers.peer1.Distributor.pharma-network.com" Created
[+] Running 29/30s peer0.Distributor.pharma-network.com
# Network pharma-network
# Volume "dockers.orderer.pharma-network.com" Created
# Volume "dockers.peer0.Consumer.pharma-network.com" Created
# Volume "dockers.peer1.Retailer.pharma-network.com" Created
# Volume "dockers.peer0.Manufacturer.pharma-network.com" Created
# Volume "dockers.peer1.Consumer.pharma-network.com" Created
# Volume "dockers.peer1.Distributor.pharma-network.com" Created
[+] Running 29/30s peer1.Distributor.pharma-network.com
# Network pharma-network
# Volume "dockers.orderer.pharma-network.com" Created
# Volume "dockers.peer0.Consumer.pharma-network.com" Created
# Volume "dockers.peer1.Retailer.pharma-network.com" Created
# Volume "dockers.peer0.Manufacturer.pharma-network.com" Created
# Volume "dockers.peer1.Consumer.pharma-network.com" Created
# Volume "dockers.peer1.Distributor.pharma-network.com" Created
[+] Running 30/30s peer0.Distributor.pharma-network.com
# Network pharma-network
# Volume "dockers.orderer.pharma-network.com" Created
# Volume "dockers.peer0.Consumer.pharma-network.com" Created
# Volume "dockers.peer1.Retailer.pharma-network.com" Created
# Volume "dockers.peer0.Manufacturer.pharma-network.com" Created
# Volume "dockers.peer1.Transporter.pharma-network.com" Created
# Volume "dockers.peer0.Transporter.pharma-network.com" Created
# Container ca_Manufacturer Started
# Container ca_Orderer Started
# Container ca_Distributor Started
# Container ca_Transporter Started
# Container peer1_Retailer Started
# Container peer1_Consumer Started
# Container orderer.pharma-network.com Started
# Container peer0.Distributor.pharma-network.com Started
# Container peer1.Consumer.pharma-network.com Started
# Container peer0.Transporter.pharma-network.com Started
# Container peer0.Manufacturer.pharma-network.com Started
# Container peer1.Transporter.pharma-network.com Started
# Container peer0.Retailer.pharma-network.com Started
# Container peer1.Manufacturer.pharma-network.com Started
# Container peer0.Consumer.pharma-network.com Started
# Container peer1.Retailer.pharma-network.com Started
# Container peer1.Distributor.pharma-network.com Started
# Container cli Started
===== Creating Channel on Orderer =====
2023-08-09 15:31:27.740 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-08-09 15:31:27.764 IST 0002 INFO [cli.common] readBlock -> Received block: 0
===== Setting Block File =====
===== Joining peer1.Manufacturer =====
2023-08-09 15:31:27.787 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-08-09 15:31:27.807 IST 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
===== Joinning peer1 of Manufacturer =====

```


5. Starting the Client Side REST API application

To start the Client Side Rest API application, navigate to the path, {Path_to_Project}/pharma-network/application and run the command : **npm start**

This will bring up the Client side REST API application.

This application can be accessed in the port 3000

Please find the screenshot below :-

```
o soumik@soumikPC:~/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/application$ npm start
Debugger listening on ws://127.0.0.1:40247/6596324-37b2-4999-81ae-ffc506b88287
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.

> pharma_network@0.0.0 start
> nodemon index.js

Debugger listening on ws://127.0.0.1:39239/e0ea41a6-4510-46ec-8395-29cc7771f62e
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching paths: ***!
[nodemon] watching extensions: *,json
[nodemon] starting 'node index.js'
[nodemon] 0
Debugger listening on ws://127.0.0.1:33981/6dd2cab6-ca2c-47ea-ab6f-13719f1f940a
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Pharma Network is running on port 3000
|
```

6. Running the Test Cases

This project contains a postman collection which contains all the necessary details to access the client side rest api and in turn interact with the underlying fabric network.

To postman collections can be found under the folder, {Path_to_Project}/pharma-network/PostMan-Collections

Please import this collection into postman.

Note, Output of the responses can be see from both the POSTMAN as well as in the terminal/console.

The following step will provide a walkthrough on the Test Case execution :-

- Open the Setup Wallet Collection and execute the API call inside. Please note that we need to change the path of the following keys to the path present on the current system for each of the calls.

For Consumer Wallet Setup

- **certificatePath:**{Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Consumer.pharma-network.com/users/Admin@Consumer.pharma-network.com/msp/signcerts/Admin@Consumer.pharma-network.com-cert.pem
- **privateKeyFilePath:**{Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Consumer.pharma-network.com/users/Admin@Consumer.pharma-network.com/msp/keystore/priv_sk

Screenshots : Postman

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, and Monitors. The main area is titled "Setup Wallet / Consumer Wallet Setup" and shows a POST request to "localhost:3000/addToWallet". The "Body" tab is selected, showing form-data parameters: "certificatePath" (value: "/home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/t"), "privateKeyFilePath" (value: "/home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/t"), and "organisationRole" (value: "Consumer"). Below the body, the response status is 200 OK with a time of 15 ms and a size of 332 B. The response body is JSON:

```
1 {  
2   "status": "success",  
3   "message": "User credentials added to wallet"  
4 }
```

Screenshots : Console

```
soumik@SoumikPC:~/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/application$ npm start  
Debugger listening on ws://127.0.0.1:40247/659b6324-37b2-4999-81ae-ffc506b88287  
For help, see: https://nodejs.org/en/docs/inspector  
Debugger attached.  
  
> pharma_network@1.0.0 start  
> nodemon index.js  
  
Debugger listening on ws://127.0.0.1:39239/e0ea41a6-4510-46ec-8395-29cc7771f62e  
For help, see: https://nodejs.org/en/docs/inspector  
Debugger attached.  
[nodemon] 3.0.1  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: js,mjs,cjs,json  
[nodemon] starting `node index.js`  
Debugger listening on ws://127.0.0.1:33981/6dd2cab6-ca2c-47ea-ab6f-13719f1f940a  
For help, see: https://nodejs.org/en/docs/inspector  
Debugger attached.  
Pharma Network is running on port 3000  
Successfully added user to wallet
```

For Retailer Wallet Setup

- **certificatePath**: {Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Retailer.pharma-network.com/users/Admin@Retailer.pharma-network.com/msp/signcerts/Admin@Retailer.pharma-network.com-cert.pem
- **privateKeyFilePath**: {Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Retailer.pharma-network.com/users/Admin@Retailer.pharma-network.com/msp/keystore/priv_sk

Screenshots : Postman

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main area shows a 'Scratch Pad' with a 'POST' request titled 'Retailer Wallet Setup'. The URL is 'localhost:3000/addToWallet'. The 'Body' tab is selected, showing 'x-www-form-urlencoded' data with three fields: 'certificatePath' (value: '/home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/'), 'privateKeyFilePath' (value: '/home/soumik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/'), and 'organisationRole' (value: 'Retailer'). Below the body, the response status is 200 OK with a message: "status": "success", "message": "User credentials added to wallet".

Screenshots : Console

```
soumik@SoumikPC:~/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/application$ npm start
Debugger listening on ws://127.0.0.1:40247/659b6324-37b2-4999-81ae-ffc506b88287
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.

> pharma_network@1.0.0 start
> nodemon index.js

Debugger listening on ws://127.0.0.1:39239/e0ea41a6-4510-46ec-8395-29cc7771f62e
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node index.js'
Debugger listening on ws://127.0.0.1:33981/6dd2cab6-ca2c-47ea-ab6f-13719f1f940a
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Pharma Network is running on port 3000
Successfully added user to wallet
Successfully added user to wallet
[]
```

For Manufacturer Wallet Setup

- **certificatePath:**{Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Manufacturer.pharma-network.com/users/Admin@Manufacturer.pharma-network.com/msp/signcerts/Admin@Manufacturer.pharma-network.com-cert.pem
- **privateKeyFilePath:**{Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Manufacturer.pharma-network.com/users/Admin@Manufacturer.pharma-network.com/msp/keystore/priv_sk

Screenshots :Postman

Working locally in Scratch Pad. Switch to a Workspace

POST Manufacturer Wallet Setup

localhost:3000/addToWallet

Key	Value	Description
<input checked="" type="checkbox"/> certificatePath	/home/sounik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/t	
<input checked="" type="checkbox"/> privateKeyFilePath	/home/sounik/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/t	
<input checked="" type="checkbox"/> organisationRole	Manufacturer	

Status: 200 OK Time: 5 ms Size: 332 B Save Response

Screenshots :Console

```
sounik@sounikPC:~/All_DEV/BlockChain/HyperLedger-Fabric/pharma-network/application$ npm start
Debugger listening on ws://127.0.0.1:40247/659b6324-37b2-4999-81ae-ff506b88287
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.

> pharma-network@1.0.0 start
> nodemon index.js

Debugger listening on ws://127.0.0.1:39239/e0ea41a6-4510-46ec-8395-29cc7771ff62e
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching paths: .+*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Debugger listening on ws://127.0.0.1:33981/6dd2cab6-ca2c-47ea-ab6f-13719f1f940a
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Pharma Network is running on port 3000
Successfully added user to wallet
Successfully added user to wallet
Successfully added user to wallet
[]
```

For Distributor Wallet Setup

- **certificatePath:**{Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Distributor.pharma-network.com/users/Admin@Distributor.pharma-network.com/msp/signcerts/Admin@Distributor.pharma-network.com-cert.pem
- **privateKeyFilePath:**{Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Distributor.pharma-network.com/users/Admin@Distributor.pharma-network.com/msp/keystore/prv_sk

Screenshots : Postman

Working locally in Scratch Pad. Switch to a Workspace

POST Distributor Wallet Setup

localhost:3000/addToWallet

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> certificatePath	/home/sounik/AI_DEV/BlockChain/HyperLedger-Fabric/pharma-network/t	
<input checked="" type="checkbox"/> privateKeyFilePath	/home/sounik/AI_DEV/BlockChain/HyperLedger-Fabric/pharma-network/t	
<input checked="" type="checkbox"/> organisationRole	Distributor	
Key	Value	Description

Status: 200 OK Time: 4 ms Size: 332 B Save Response

Screenshots : Console

```

Debugger listening on ws://127.0.0.1:40247/659b6324-37b2-4999-81ae-ffc506b88287
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.

> pharma_network@1.0.0 start
> nodemon index.js
Debugger listening on ws://127.0.0.1:39239/e0ea41a6-4510-46ec-8395-29cc7771f62e
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node index.js'
Debugger listening on ws://127.0.0.1:33981/6dd2cab6-ca2c-47ea-ab6f-13719ff940a
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Pharma Network is running on port 3000
Successfully added user to wallet

```

For Transporter Wallet Setup

- **certificatePath**: {Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Transporter.pharma-network.com/users/Admin@Transporter.pharma-network.com/msp/signcerts/Admin@Transporter.pharma-network.com-cert.pem
- **privateKeyFilePath**: {Path_to_Project}/pharma-network/test-network-man/organizations/peerOrganizations/Transporter.pharma-network.com/users/Admin@Transporter.pharma-network.com/msp/keystore/prv_sk

Screenshots : Postman

Postman interface showing a successful POST request to 'localhost:3001/addToWallet' for 'Transporter Wallet Setup'. The request body includes parameters: certificatePath, privateKeyFilePath, and organisationRole. The response is a JSON object with 'status': 'success' and 'message': 'User credentials added to wallet'.

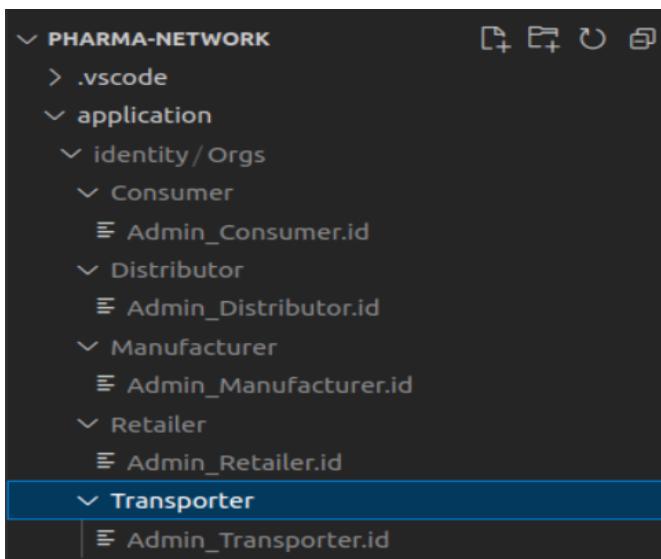
Screenshots : Console

```
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.

> pharma_network@1.0.0 start
> nodemon index.js

Debugger listening on ws://127.0.0.1:39239/e0ea41a6-4510-46ec-8395-29cc7771f62e
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js` in /home/soumik/AI_DEV/BlockChain/HyperLedger-Fabric/pharma-network
Debugger listening on ws://127.0.0.1:3981/6dd2cab6-ca2c-47ea-ab6f-13719f1f940a
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Pharma Network is running on port 3000
Successfully added user to wallet
[]
```

Once all the wallet transactions are completed, you can view the identity files created in the system under the folder {Path_to_Project}/pharma-network/application/identity/Orgs



- Once the wallet creation is completed, we can move forward to register the companies and the drug. Open the Collection Initiation. This collection contains all the API call to register company and drug

Create Company : Manufacturer

Screenshots : Postman

The screenshot shows the Postman interface with a POST request to 'localhost:3000/registerCompany'. The request body is defined with the following parameters:

Key	Value	Description
companyCRN	MAN001	
companyName	Sun Pharma	
location	Chennai	
organisationRole	MANUFACTURER	

The response body is a JSON object:

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
{
  "status": "success",
  "message": "Company Added Successfully",
  "output": {
    "name": "MAN001-Sun Pharma",
    "name": "Sun Pharma",
    "location": "Chennai",
    "organisationRole": "MANUFACTURER",
    "hierarchyKey": 1,
    "createdAt": {
      "nanos": 314000000,
      "seconds": 1691579382,
      "low": 1691579382,
      "high": 0,
      "unsigned": true
    },
    "updatedAt": {
      "nanos": 314000000,
      "seconds": 1691579382,
      "low": 1691579382,
      "high": 0,
      "unsigned": true
    }
  }
}
  
```

Screenshots : Console

For help, see: <https://nodejs.org/en/docs/inspector>
Debugger attached.

Pharma Network is running on port 3000
Successfully added user to wallet
Successfully added user to wallet

```
{
  companyID: 'MAN001-Sun Pharma',
  name: 'Sun Pharma',
  location: 'Chennai',
  organisationRole: 'MANUFACTURER',
  hierarchyKey: 1,
  createdAt: {
    nanos: 314000000,
    seconds: { low: 1691579382, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 314000000,
    seconds: { low: 1691579382, high: 0, unsigned: true }
  }
}
```

Create Company : Distributor

Screenshots : Postman

The screenshot shows the Postman application interface. A POST request is being made to the endpoint `localhost:3000/registerCompany`. The request body is set to `www-form-urlencoded` and contains the following parameters:

KEY	VALUE	DESCRIPTION
companyCRN	DIST001	
CompanyName	VG Pharma	
Location	Vizag	
organisationRole	Distributor	

The response status is 200 OK, with a time of 2.13 s and a size of 631 B. The response body is a JSON object:

```
1
2     "status": "success",
3     "message": "Company Added Successfully",
4     "output": {
5       "companyID": "DIST001-VG Pharma",
6       "name": "VG Pharma",
7       "location": "Vizag",
8       "organisationRole": "DISTRIBUTOR",
9       "hierarchyKey": 2,
10      "createdAt": {
11        "nanos": 890000000,
12        "seconds": {
13          "low": 1691579579,
14          "high": 0,
15          "unsigned": true
16        }
17      },
18      "updatedAt": {
19        "nanos": 890000000,
20        "seconds": {
21          "low": 1691579579,
22          "high": 0,
23          "unsigned": true
24        }
25      }
}
```

Screenshots : Console

```
{
  companyID: 'DIST001-VG Pharma',
  name: 'VG Pharma',
  location: 'Vizag',
  organisationRole: 'DISTRIBUTOR',
  hierarchyKey: 2,
  createdAt: {
    nanos: 890000000,
    seconds: { low: 1691579579, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 890000000,
    seconds: { low: 1691579579, high: 0, unsigned: true }
  }
}
```

Create Company : Transporter – FedEx

Screenshots : Postman

The screenshot shows the Postman interface with a successful API call. The URL is `localhost:3000/registerCompany`. The request body contains the following JSON:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
{
  "status": "success",
  "message": "Company Added Successfully",
  "data": {
    "companyID": "TRA001-FedEx",
    "name": "FedEx",
    "location": "Delhi",
    "organisationRole": "TRANSPORTER",
    "hierarchyKey": 4,
    "createdAt": {
      "nanos": 588000000,
      "seconds": { "low": 1691579837, "high": 0, "unsigned": true }
    },
    "updatedAt": {
      "nanos": 588000000,
      "seconds": { "low": 1691579837, "high": 0, "unsigned": true }
    }
  }
}
```

The response status is 200 OK with a time of 217 ms and a size of 622 B.

Screenshots : Console

```
{
  companyID: 'TRA001-FedEx',
  name: 'FedEx',
  location: 'Delhi',
  organisationRole: 'TRANSPORTER',
  hierarchyKey: 4,
  createdAt: {
    nanos: 588000000,
    seconds: { low: 1691579837, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 588000000,
    seconds: { low: 1691579837, high: 0, unsigned: true }
  }
}
```

Create Company : Transporter – Blue Dart

Screenshots : Postman

The screenshot shows the Postman application interface. A POST request is being made to the endpoint `localhost:3000/registerCompany`. The request body contains the following JSON data:

```
POST /registerCompany : Test
Initiation / Register Company : Transporter Blue-Dart
localhost:3000/registerCompany
Params Authorization Headers (B) Body * Pre-request Script Tests * Settings
@none @form-data @www-form-urlencoded @raw @binary @GraphQL
KEY VALUE DESCRIPTION
companyCRN TRA002
companyName Blue Dart
Location Bangalore
organisationRole Transporter
```

The response status is 200 OK, and the response body is:

```
Pretty Raw Preview Visualize JSON
1
2   "status": "success",
3   "message": "Company Added Successfully",
4   "output": [
5     {
6       "companyID": "TRA002-Blue Dart",
7       "name": "Blue Dart",
8       "location": "Bangalore",
9       "organisationRole": "TRANSPORTER",
10      "hierarchyKey": 4,
11      "createdAt": {
12        "nanos": 180000000,
13        "seconds": { "low": 1691579891, "high": 0, "unsigned": true }
14      },
15      "updatedAt": {
16        "nanos": 180000000,
17        "seconds": { "low": 1691579891, "high": 0, "unsigned": true }
18      }
19    }
20  ]
```

Screenshots : Console

```
}
```

```
{
  companyID: 'TRA002-Blue Dart',
  name: 'Blue Dart',
  location: 'Bangalore',
  organisationRole: 'TRANSPORTER',
  hierarchyKey: 4,
  createdAt: {
    nanos: 180000000,
    seconds: { low: 1691579891, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 180000000,
    seconds: { low: 1691579891, high: 0, unsigned: true }
  }
}
```

Create Company : Retailer

Screenshots : Postman

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (History, API), 'Environments', 'Mock Services', 'Monitors', and 'History'. The main area has a 'Scratch Pad' tab selected. A 'POST Register Company : Retailer' request is being edited. The 'Body' tab is active, showing a JSON payload:

```
POST /registerCompany
{
  "companyID": "RET002-Upgrad",
  "name": "Upgrad",
  "location": "Mumbai",
  "organisationRole": "RETAILER",
  "hierarchyKey": 3,
  "createdAt": {
    "nanos": 55000000,
    "seconds": { "low": 1691579929, "high": 0, "unsigned": true }
  },
  "updatedAt": {
    "nanos": 55000000,
    "seconds": { "low": 1691579929, "high": 0, "unsigned": true }
  }
}
```

The 'Body' tab also includes tabs for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. Below the body editor, the response status is shown as 'Status: 200 OK'.

Screenshots : Console

A terminal window displays a JSON object. The object represents a company registration response. It includes fields like companyID, name, location, organisationRole, hierarchyKey, createdAt, and updatedAt, all containing specific values such as 'RET002-Upgrad', 'Upgrad', 'Mumbai', 'RETAILER', and timestamp details.

```
{
  "companyID": "RET002-Upgrad",
  "name": "Upgrad",
  "location": "Mumbai",
  "organisationRole": "RETAILER",
  "hierarchyKey": 3,
  "createdAt": {
    "nanos": 55000000,
    "seconds": { "low": 1691579929, "high": 0, "unsigned": true }
  },
  "updatedAt": {
    "nanos": 55000000,
    "seconds": { "low": 1691579929, "high": 0, "unsigned": true }
  }
}
```

Add Drug: 001

Screenshot: Postman

The screenshot shows the Postman application interface. The top navigation bar includes File, Edit, View, Help, Home, Workspaces, and Explore. A search bar at the top right says "Working locally in Scratch Pad. Switch to a Workspace". The main area shows a collection named "Initiation" with several items like "Register Company - Manufacturer", "Add Drug - Paracetamol - 001", etc. A POST request is selected with the URL "localhost:3000/addDrug". The "Body" tab is active, showing a JSON payload:

```
1 {
2     "status": "success",
3     "message": "Drug Added Successfully",
4     "output": [
5         {
6             "productID": "Paracetamol-001",
7             "name": "Paracetamol",
8             "manufacturer": "MAN001-Sun Pharma",
9             "manufacturingDate": "1/1/2020",
10            "expiryDate": "1/1/2023",
11            "owner": "MAN001-Sun Pharma",
12            "shipment": '',
13            "createdAt": {
14                "nanos": 493000000,
15                "seconds": { "low": 1691580155, "high": 0, "unsigned": true }
16            },
17            "updatedAt": {
18                "nanos": 493000000,
19                "seconds": { "low": 1691580155, "high": 0, "unsigned": true }
20            }
21        }
22    ]
23 }
```

The status bar at the bottom indicates "Status: 200 OK Time: 2.11 s Size: 691 B Save Response".

Screenshot: Console

```
{
  productID: 'Paracetamol-001',
  name: 'Paracetamol',
  manufacturer: 'MAN001-Sun Pharma',
  manufacturingDate: '1/1/2020',
  expiryDate: '1/1/2023',
  owner: 'MAN001-Sun Pharma',
  shipment: '',
  createdAt: {
    nanos: 493000000,
    seconds: { low: 1691580155, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 493000000,
    seconds: { low: 1691580155, high: 0, unsigned: true }
  }
}
```

Add Drug: 002

Screenshot: Postman

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like 'Scratch Pad', 'Collections', 'Allu', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main area shows a 'POST Add Drug : Paracetamol' request. The 'Body' tab is selected, showing a JSON payload:

```
1
2   "status": "success",
3   "message": "Drug Added Successfully",
4   "drug": {
5     "productID": "Paracetamol-002",
6     "name": "Paracetamol",
7     "manufacturer": "MAN001-Sun Pharma",
8     "manufacturingDate": "1/1/2020",
9     "expiryDate": "1/1/2023",
10    "owner": "MAN001-Sun Pharma",
11    "shipment": '',
12    "createdAt": {
13      "nanos": 583000000,
14      "seconds": {
15        "low": 1691580220,
16        "high": 0,
17        "unsigned": true
18      }
19    },
20    "updatedAt": {
21      "nanos": 583000000,
22      "seconds": {
23        "low": 1691580220,
24        "high": 0,
25        "unsigned": true
26      }
27    }
28  }
```

The status bar at the bottom indicates 'Status: 200 OK Time: 212 s Size: 691 B Save Response'.

Screenshot: Console

```
{
  productID: 'Paracetamol-002',
  name: 'Paracetamol',
  manufacturer: 'MAN001-Sun Pharma',
  manufacturingDate: '1/1/2020',
  expiryDate: '1/1/2023',
  owner: 'MAN001-Sun Pharma',
  shipment: '',
  createdAt: {
    nanos: 583000000,
    seconds: { low: 1691580220, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 583000000,
    seconds: { low: 1691580220, high: 0, unsigned: true }
  }
}
```

Add Drug: 003

Screenshot: Postman

The screenshot shows the Postman interface with a successful API call. The URL is `localhost:3000/addDrug`. The request body is a JSON object with fields: `drugName`, `serialNo`, `mfgDate`, `expDate`, `companyCRN`, and `organisationRole`. The response status is 200 OK, and the response body is a JSON object indicating success.

```

{
  "status": "success",
  "message": "Drug Added Successfully",
  "owner": "MAN001-Sun Pharma",
  "productID": "Paracetamol-003",
  "name": "Paracetamol",
  "manufacturer": "MAN001-Sun Pharma",
  "manufacturingDate": "1/1/2020",
  "expiryDate": "1/1/2023",
  "owner": "MAN001-Sun Pharma",
  "shipment": "",
  "createdAt": {
    "nanos": 756000000,
    "seconds": { "low": 1691580245, "high": 0, "unsigned": true }
  },
  "updatedAt": {
    "nanos": 756000000,
    "seconds": { "low": 1691580245, "high": 0, "unsigned": true }
  }
}

```

Screenshot: Console

```
{
  productID: 'Paracetamol-003',
  name: 'Paracetamol',
  manufacturer: 'MAN001-Sun Pharma',
  manufacturingDate: '1/1/2020',
  expiryDate: '1/1/2023',
  owner: 'MAN001-Sun Pharma',
  shipment: '',
  createdAt: {
    nanos: 756000000,
    seconds: { low: 1691580245, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 756000000,
    seconds: { low: 1691580245, high: 0, unsigned: true }
  }
}
```

Add Drug: 004

Screenshot: Postman

The screenshot shows the Postman application interface. On the left, there's a sidebar with collections like 'Scratch Pad', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main area shows a 'POST' request to 'localhost:3000/addDrug'. The 'Body' tab is selected, showing a JSON payload:

```

{
  "drugName": "Paracetamol",
  "serialNo": "004",
  "mfgDate": "1/1/2020",
  "expDate": "1/1/2023",
  "companyCRN": "MAN001-Sun Pharma",
  "organisationRole": "Manufacturer"
}

```

The response status is 200 OK.

Screenshot: Console

```
{
  productID: 'Paracetamol-004',
  name: 'Paracetamol',
  manufacturer: 'MAN001-Sun Pharma',
  manufacturingDate: '1/1/2020',
  expiryDate: '1/1/2023',
  owner: 'MAN001-Sun Pharma',
  shipment: '',
  createdAt: {
    nanos: 902000000,
    seconds: { low: 1691580260, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 902000000,
    seconds: { low: 1691580260, high: 0, unsigned: true }
  }
}
```

- Once the drug creation is completed we can move forward to perform the supply chain Test Case. The supply chain TC are split into three parts. Open the Supply Chain collection to access the Rest API Calls

● PART A

Create PO : Distributor to Manufacturer

Postman :

File Edit View Help

Home Workspaces Explore

Scratch Pad New Import POST Create PO + ***

Working locally in Scratch Pad. Switch to a Workspace

No Environment

Save Cookies Send

Collections

Artis

Environments

Mock Servers

Monitors

History

History Initiation Setup Wallet Supply Chain Part A Part B Part C

POST /localhost:3000/createPO

POST Body Params Authorization Headers (B) Body Pre-request Script Tests Settings

None Form-data WWW-Form-URLencoded Raw Binary GraphQL

KEY VALUE DESCRIPTION

buyerCRN DIST001-VG Pharma

sellerCRN MAN001-Sun Pharma

drugName Paracetamol

quantity 3

organisationRole Distributor

Key Value Description

Body Cookies Headers (B) Test Results (1/1)

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 2.12 s Size: 644 B Save Response

```
1 {
2     "status": "success",
3     "message": "PO created Successfully",
4     "output": {
5         "poID": "DIST001-VG Pharma-Paracetamol",
6         "drugName": "Paracetamol",
7         "quantity": "3",
8         "buyer": "DIST001-VG Pharma",
9         "seller": "MAN001-Sun Pharma",
10        "createAt": {
11            "nanos": 293000000,
12            "seconds": 16915808846,
13            "low": 16915808846,
14            "high": 0,
15            "unsigned": true
16        }
17    },
18    "updatedAt": {
19        "nanos": 293000000,
20        "seconds": 16915808846,
21        "low": 16915808846,
22        "high": 0,
23        "unsigned": true
24    }
25 }
```

Find and Replace Console Runner Trash

Console:

```
{  
  poID: 'DIST001-VG Pharma-Paracetamol',  
  drugName: 'Paracetamol',  
  quantity: '3',  
  buyer: 'DIST001-VG Pharma',  
  seller: 'MAN001-Sun Pharma',  
  createdAt: {  
    nanos: 293000000,  
    seconds: { low: 1691580846, high: 0, unsigned: true }  
  },  
  updatedAt: {  
    nanos: 293000000,  
    seconds: { low: 1691580846, high: 0, unsigned: true }  
  }  
}
```

Create Shipment : Distributor to Manufacturer

Postman :

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Scratch Pad' selected, showing a tree structure with 'Part A' expanded, containing 'Create PO', 'Create Shipment', and 'Update Shipment'. The main area shows a 'POST Create Shipment' request to 'localhost:3000/createShipment'. The 'Body' tab is selected, showing a JSON payload:

```

{
  "buyerCRN": "DIST001-VG Pharma",
  "drugName": "Paracetamol",
  "listOfAssets": ["001","002","003"],
  "transporterCRN": "TRA001-FedEx",
  "organisationRole": "Manufacturer"
}

```

The response tab shows a successful 200 OK status with the message: "Shipment created Successfully".

Console :

```

{
  "shipmentID": "DIST001-VG Pharma-Paracetamol",
  "creator": "x509:/C=US/ST=California/L=San Francisco/O=admin/CN=Admin@Manufacturer.pharma-network.com:/C=US/ST=California/L=San Francisco/O=Manufacturer.pharma-network.com/CN=ca.Manufacturer.pharma-network.com",
  "assets": [ 'Paracetamol-001', 'Paracetamol-002', 'Paracetamol-003' ],
  "transporter": 'TRA001-FedEx',
  "status": 'in-transit',
  "createdAt": {
    "nanos": 980000000,
    "seconds": 1691580865,
    "low": 1691580865,
    "high": 0,
    "unsigned": true
  },
  "updatedAt": {
    "nanos": 980000000,
    "seconds": { low: 1691580865, high: 0, unsigned: true }
  }
}

```

Create Shipment : Update Shipment

Postman :

The screenshot shows the Postman application interface. On the left, the 'Scratch Pad' sidebar lists collections, environments, mock servers, monitors, and history. In the center, a 'POST Update Shipment' request is being edited. The 'Body' tab is selected, showing a JSON payload with fields: buyerCRN (DIST001-VG Pharma), drugName (Paracetamol), transporterCRN (TRA001-FedEx), and organisationRole (Transporter). The 'Pretty' tab shows the JSON structure. At the bottom, the response status is 200 OK, time 2.12s, and size 887B.

Console

```
{
  "shipmentID": "DIST001-VG Pharma-Paracetamol",
  "creator": "x509::C=US/ST=California/L=San Francisco/OU=admin/CN=Admin@Manufacturer.pharma-network.com::C=US/ST=California/L=San Francisco/O=Manufacturer.pharma-network.com/CN=ca.Manufacturer.pharma-network.com",
  "assets": [ 'Paracetamol-001', 'Paracetamol-002', 'Paracetamol-003' ],
  "transporter": 'TRA001-FedEx',
  "status": 'Delivered',
  "createdAt": {
    "nanos": 880000000,
    "seconds": { low: 1691580865, high: 0, unsigned: true }
  },
  "updatedAt": {
    "nanos": 166000000,
    "seconds": { low: 1691581056, high: 0, unsigned: true }
  }
}
```

- Part B

Create PO : Retailer to Distributor

Postman :

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (Initiation, Setup Wallet, Supply Chain), 'Environments', 'Most Servers', 'Monitors', and 'History'. The main area shows a 'Scratch Pad' with a 'POST Create PO' collection. Underneath, a 'POST' request is being made to 'localhost:3000/createPO'. The 'Body' tab is selected, showing the following JSON payload:

```

{
  "buyerCRN": "RET002-Upgrad",
  "sellerCRN": "DIST001-VG Pharma",
  "drugName": "Paracetamol",
  "quantity": 2,
  "organisationRole": "Retailer"
}

```

The response status is 200 OK, and the response body is displayed in the 'Pretty' tab:

```

{
  "status": "success",
  "message": "PO created Successfully",
  "output": {
    "poID": "RET002-Upgrad-Paracetamol",
    "drugName": "Paracetamol",
    "organisationRole": "Retailer",
    "buyer": "RET002-Upgrad",
    "seller": "DIST001-VG Pharma",
    "createdAt": {
      "nanos": 479000000,
      "seconds": 1691581424,
      "low": 1691581424,
      "high": 0,
      "unsigned": true
    },
    "updatedAt": {
      "nanos": 479000000,
      "seconds": 1691581424,
      "low": 1691581424,
      "high": 0,
      "unsigned": true
    }
  }
}

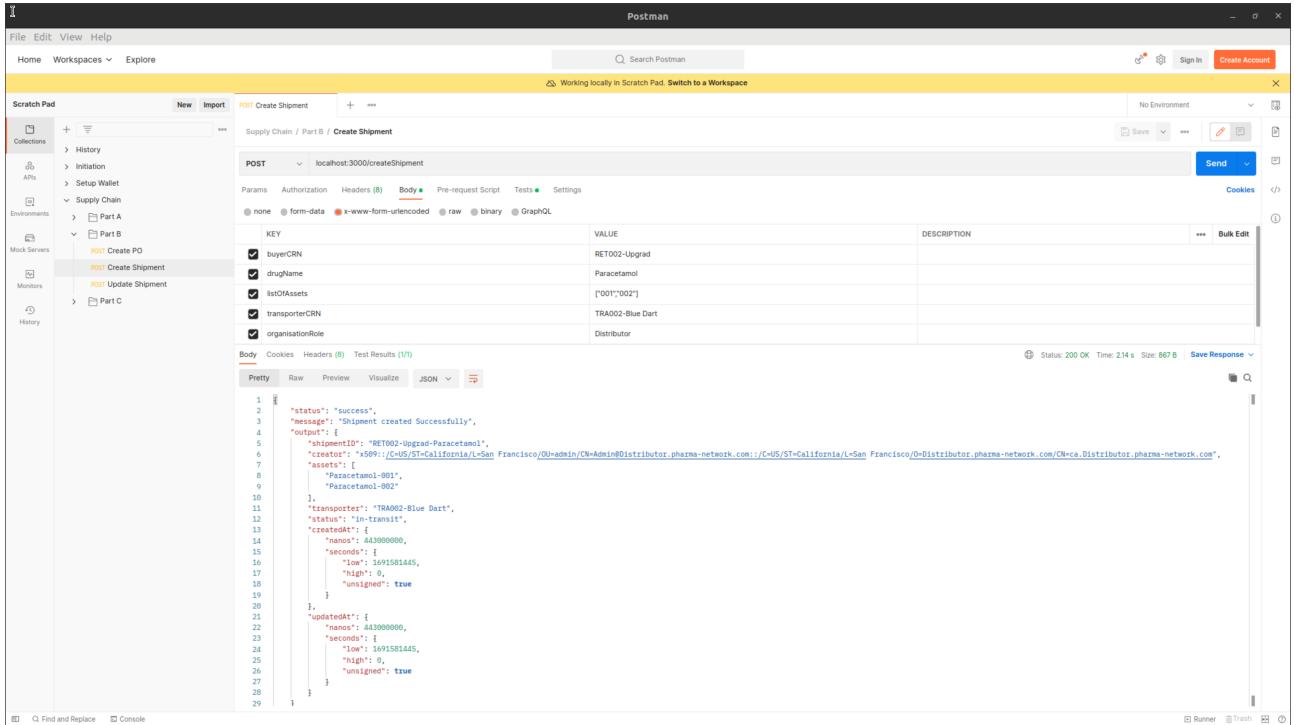
```

Console:

```
{
  "poID": "RET002-Upgrad-Paracetamol",
  "drugName": "Paracetamol",
  "quantity": 2,
  "buyer": "RET002-Upgrad",
  "seller": "DIST001-VG Pharma",
  "createdAt": {
    "nanos": 479000000,
    "seconds": { low: 1691581424, high: 0, unsigned: true }
  },
  "updatedAt": {
    "nanos": 479000000,
    "seconds": { low: 1691581424, high: 0, unsigned: true }
  }
}
```

Create Shipment : Distributor to Retailer

Postman :



The screenshot shows the Postman application interface. A POST request is being made to `localhost:3000/createShipment`. The request body is defined with the following parameters:

KEY	VALUE	DESCRIPTION
buyerCRN	RET002-Upgrad	
drugname	Paracetamol	
listOfAssets	["001","002"]	
transporterCRN	TRA002-Blue Dart	
organisationRole	Distributor	

The response status is 200 OK, and the response body is displayed in JSON format:

```
1  {
2   "status": "Success",
3   "message": "Shipment created Successfully",
4   "output": [
5     {
6       "shipmentID": "RET002-Upgrad-Paracetamol",
7       "creator": "x509::/C=US/ST=California/L=San Francisco/O=admin@Distributor.pharma-network.com::C=US/ST=California/L=San Francisco/O=Distributor.pharma-network.com/CN=ca.Distributor.pharma-network.com",
8       "assets": [
9         "Paracetamol-001",
10        "Paracetamol-002"
11      ],
12      "transporter": "TRA002-Blue Dart",
13      "status": "in-transit",
14      "createdAt": {
15        "nanos": 443000000,
16        "seconds": 1691581445,
17        "low": 1691581445,
18        "high": 0,
19        "unsigned": true
20      },
21      "updatedAt": {
22        "nanos": 443000000,
23        "seconds": {
24          "low": 1691581445,
25          "high": 0,
26        "unsigned": true
27      }
28    }
29  }
```

Console :

```
{
  shipmentID: 'RET002-Upgrad-Paracetamol',
  creator: 'x509::/C=US/ST=California/L=San Francisco/O=admin@Distributor.pharma-network.com::C=US/ST=California/L=San Francisco/O=Distributor.pharma-network.com/CN=ca.Distributor.pharma-network.com',
  assets: [ 'Paracetamol-001', 'Paracetamol-002' ],
  transporter: 'TRA002-Blue Dart',
  status: 'in-transit',
  createdAt: {
    nanos: 443000000,
    seconds: { low: 1691581445, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 443000000,
    seconds: { low: 1691581445, high: 0, unsigned: true }
  }
}
```

Update Shipment

Postman :

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (History, Initiation, Setup Wallet, Supply Chain), 'Environments' (Mock Servers, Monitors, History), and a 'Scratch Pad' section. In the center, a 'POST Update Shipment' request is being made to 'localhost:3000/updateShipment'. The 'Body' tab is selected, showing a JSON payload with fields: buyerCRN (RETO02-Upgrad), drugname (Paracetamol), transporterCRN (TRA002-Blue Dart), and organisationRole (Transporter). The response status is 200 OK, time 213 s, size 868 B. The JSON response body is displayed below:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

"status": "success",
"message": "Shipment created Successfully",
"output": {
"shipmentID": "RETO02-upgrad-Paracetamol",
"creator": "x509:/C=US/ST=California/L=San Francisco/O=admin/CN=Admin@Distributor.pharma-network.com::/C=US/ST=California/L=San Francisco/O=Distributor.pharma-network.com/CN=ca.Distributor.pharma-network.com",
"assets": [
{"assetID": "Paracetamol-001",
"name": "Paracetamol",
"status": "Delivered",
"createdAt": {
"name": "2024-01-15T00:00:00Z",
"seconds": {
"low": 1691581445,
"high": 0,
"unsigned": true
},
"updatedAt": {
"name": "2024-01-15T00:00:00Z",
"seconds": {
"low": 1691581480,
"high": 0,
"unsigned": true
}
},
"nanos: 443000000,
seconds: { low: 1691581445, high: 0, unsigned: true }
},
updatedAt: {
"name": "2024-01-15T00:00:00Z",
"seconds": { low: 1691581480, high: 0, unsigned: true }
}
}

Console

```
{  
  shipmentID: 'RETO02-Upgrad-Paracetamol',  
  creator: 'x509:/C=US/ST=California/L=San Francisco/O=admin/CN=Admin@Distributor.pharma-network.com::/C=US/ST=California/L=San Francisco/O=Distributor.pharma-network.com/CN=ca.Distributor.pharma-network.com',  
  assets: [ 'Paracetamol-001', 'Paracetamol-002' ],  
  transporter: 'TRA002-Blue Dart',  
  status: 'Delivered',  
  createdAt: {  
    name: '2024-01-15T00:00:00Z',  
    seconds: { low: 1691581445, high: 0, unsigned: true }  
  },  
  updatedAt: {  
    name: '2024-01-15T00:00:00Z',  
    seconds: { low: 1691581480, high: 0, unsigned: true }  
  }  
}
```

- Part C

Retail Drug : Update Shipment

Postman :

The screenshot shows the Postman application interface. On the left, there's a sidebar with collections like History, Initiation, Setup Wallet, Supply Chain, Part A, Part B, and Part C, with 'Retail Drug' selected. The main area has a 'POST Retail Drug' request. The 'Body' tab is selected, showing a form-data object with fields: drugName (Paracetamol), serialNo (001), retailerCRN (RET002-Upgrad), customerAadhar (AAD001), and organisationRole (Retailer). The response pane shows a JSON object with a 'status' field of 'success' and a 'message' field of 'Drug sold Successfully'. The JSON output is displayed in a 'Pretty' format.

```

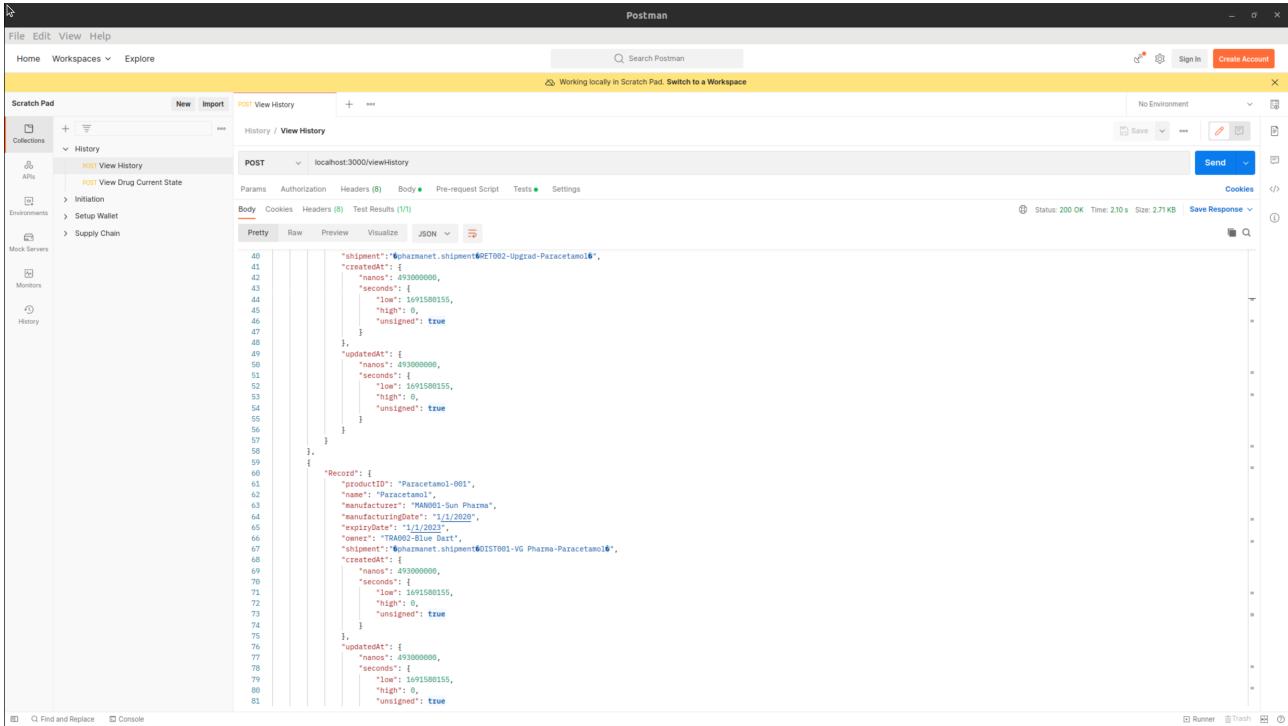
{
  "status": "success",
  "message": "Drug sold Successfully",
  "output": [
    {
      "productID": "Paracetamol-001",
      "name": "Paracetamol",
      "manufacturer": "MAN001-Sun Pharma",
      "manufacturingDate": "1/1/2020",
      "expiryDate": "1/1/2023",
      "owner": "A0001",
      "shipment": "\x00pharmanet.shipment\x00RET002-Upgrad-Paracetamol\x00",
      "createdAt": {
        "nanos": 493000000,
        "seconds": {
          "low": 1691580155,
          "high": 0,
          "unsigned": true
        }
      },
      "updatedAt": {
        "nanos": 493000000,
        "seconds": {
          "low": 1691580155,
          "high": 0,
          "unsigned": true
        }
      }
    }
  ]
}
  
```

Console

```
{
  "productID": "Paracetamol-001",
  "name": "Paracetamol",
  "manufacturer": "MAN001-Sun Pharma",
  "manufacturingDate": "1/1/2020",
  "expiryDate": "1/1/2023",
  "owner": "A0001",
  "shipment": "\x00pharmanet.shipment\x00RET002-Upgrad-Paracetamol\x00",
  "createdAt": {
    "nanos": 493000000,
    "seconds": {
      "low": 1691580155,
      "high": 0,
      "unsigned": true
    }
  },
  "updatedAt": {
    "nanos": 493000000,
    "seconds": {
      "low": 1691580155,
      "high": 0,
      "unsigned": true
    }
  }
}
```

- Once the supply chain is completed we can now view the Drug History

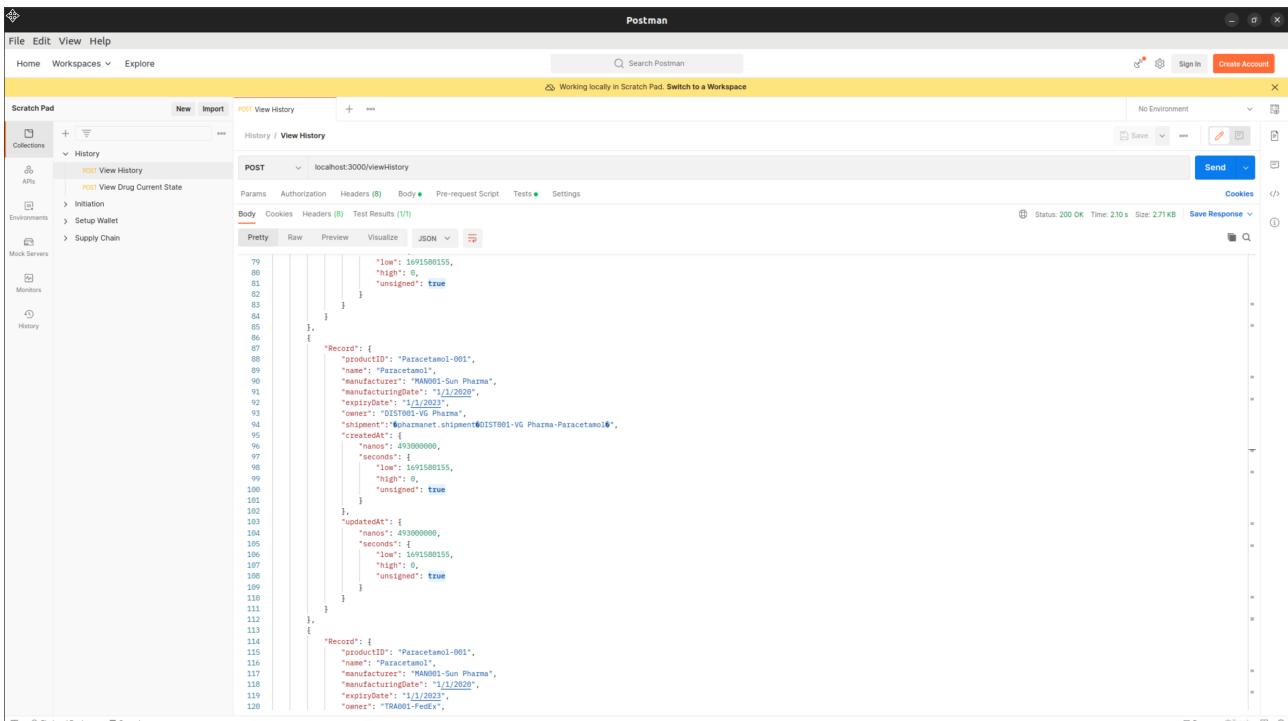
Postman :



The screenshot shows the Postman application interface. The left sidebar contains collections, environments, and a history section with items like "View History", "View Drug Current State", "Initiation", "Setup Wallet", and "Supply Chain". The main area shows a POST request to "localhost:3000/viewHistory". The "Body" tab displays a JSON payload representing a list of drug records. The response status is 200 OK, and the response body is identical to the one shown in the previous screenshot.

```

POST /localhost:3000/viewHistory
{
  "Record": [
    {
      "productID": "Paracetamol-001",
      "name": "Paracetamol",
      "manufacturer": "MAN001-Sun Pharma",
      "manufacturingDate": "1/1/2020",
      "expiryDate": "1/1/2023",
      "owner": "D0002-Blue Dart",
      "shipment": "PharmNet shipment#D01T001-VG Pharma-Paracetamol",
      "createdAt": {
        "nanos": 493000000,
        "seconds": 1691580155,
        "low": 1691580155,
        "high": 0,
        "unsigned": true
      },
      "updatedAt": {
        "nanos": 493000000,
        "seconds": 1691580155,
        "low": 1691580155,
        "high": 0,
        "unsigned": true
      }
    },
    {
      "productID": "Paracetamol-001",
      "name": "Paracetamol",
      "manufacturer": "MAN001-Sun Pharma",
      "manufacturingDate": "1/1/2020",
      "expiryDate": "1/1/2023",
      "owner": "D0002-Blue Dart",
      "shipment": "PharmNet shipment#D01T001-VG Pharma-Paracetamol",
      "createdAt": {
        "nanos": 493000000,
        "seconds": 1691580155,
        "low": 1691580155,
        "high": 0,
        "unsigned": true
      },
      "updatedAt": {
        "nanos": 493000000,
        "seconds": 1691580155,
        "low": 1691580155,
        "high": 0,
        "unsigned": true
      }
    }
  ]
}
  
```



This screenshot is nearly identical to the one above, showing the same POST request to "localhost:3000/viewHistory". The "Body" tab displays the same JSON payload of drug records. The response status is 200 OK, and the response body is identical to the one shown in the previous screenshot.

```

POST /localhost:3000/viewHistory
{
  "Record": [
    {
      "productID": "Paracetamol-001",
      "name": "Paracetamol",
      "manufacturer": "MAN001-Sun Pharma",
      "manufacturingDate": "1/1/2020",
      "expiryDate": "1/1/2023",
      "owner": "D0002-Blue Dart",
      "shipment": "PharmNet shipment#D01T001-VG Pharma-Paracetamol",
      "createdAt": {
        "nanos": 493000000,
        "seconds": 1691580155,
        "low": 1691580155,
        "high": 0,
        "unsigned": true
      },
      "updatedAt": {
        "nanos": 493000000,
        "seconds": 1691580155,
        "low": 1691580155,
        "high": 0,
        "unsigned": true
      }
    },
    {
      "productID": "Paracetamol-001",
      "name": "Paracetamol",
      "manufacturer": "MAN001-Sun Pharma",
      "manufacturingDate": "1/1/2020",
      "expiryDate": "1/1/2023",
      "owner": "D0002-Blue Dart",
      "shipment": "PharmNet shipment#D01T001-VG Pharma-Paracetamol",
      "createdAt": {
        "nanos": 493000000,
        "seconds": 1691580155,
        "low": 1691580155,
        "high": 0,
        "unsigned": true
      },
      "updatedAt": {
        "nanos": 493000000,
        "seconds": 1691580155,
        "low": 1691580155,
        "high": 0,
        "unsigned": true
      }
    }
  ]
}
  
```

Postman

File Edit View Help

Home Workspaces Explore

Scratch Pad New Import POST View History

Working locally in Scratch Pad. Switch to a Workspace

History / View History

POST localhost:3000/viewHistory

Params Authorization Headers (8) Body (1) Pre-request Script Tests Settings

Body Cookies Headers (8) Test Results (1)

Pretty Raw Preview Visualize JSON

127 },
128 "unsigned": true
129 },
130 "updatedAt": [
131 "nanos": 493000000,
132 "seconds": 1691580155,
133 "low": 1691580155,
134 "high": 0,
135 "unsigned": true
136 }
137 },
138],
139 },
140 [{
141 "Record": {
142 "productId": "Paracetamol-001",
143 "name": "Paracetamol",
144 "manufacturer": "MAMBOO-Sun Pharma",
145 "manufacturingDate": "1/1/2029",
146 "expiryDate": "1/1/2023",
147 "owner": "MAMBOO-Sun Pharma",
148 "status": "Active",
149 "createdAt": {
150 "nanos": 493000000,
151 "seconds": 1691580155,
152 "low": 1691580155,
153 "high": 0,
154 "unsigned": true
155 },
156 "updatedAt": [
157 "nanos": 493000000,
158 "seconds": 1691580155,
159 "low": 1691580155,
160 "high": 0,
161 "unsigned": true
162 }],
163 },
164],
165 },
166]
167 }
168 }

Save Send Cookies Status: 200 OK Time: 2.10 s Size: 2.71 KB Save Response

Find and Replace Console Runner Trash

Console

```
Record: {
  productID: 'Paracetamol-001',
  name: 'Paracetamol',
  manufacturer: 'MAN001-Sun Pharma',
  manufacturingDate: '1/1/2020',
  expiryDate: '1/1/2023',
  owner: 'AAD001',
  shipment: '\x00pharmanet.shipment\x00RET002-Upgrad-Paracetamol\x00',
  createdAt: [Object],
  updatedAt: [Object]
}
{
  Record: {
    productID: 'Paracetamol-001',
    name: 'Paracetamol',
    manufacturer: 'MAN001-Sun Pharma',
    manufacturingDate: '1/1/2020',
    expiryDate: '1/1/2023',
    owner: 'RET002-Upgrad',
    shipment: '\x00pharmanet.shipment\x00RET002-Upgrad-Paracetamol\x00',
    createdAt: [Object],
    updatedAt: [Object]
  }
},
{
  Record: {
    productID: 'Paracetamol-001',
    name: 'Paracetamol',
    manufacturer: 'MAN001-Sun Pharma',
    manufacturingDate: '1/1/2020',
    expiryDate: '1/1/2023',
    owner: 'TRA002-Blue Dart',
    shipment: '\x00pharmanet.shipment\x00DIST001-VG Pharma-Paracetamol\x00',
    createdAt: [Object],
    updatedAt: [Object]
  }
},
{
  Record: {
    productID: 'Paracetamol-001',
    name: 'Paracetamol',
    manufacturer: 'MAN001-Sun Pharma',
    manufacturingDate: '1/1/2020',
    expiryDate: '1/1/2023',
    owner: 'DIST001-VG Pharma',
    shipment: '\x00pharmanet.shipment\x00DIST001-VG Pharma-Paracetamol\x00',
    createdAt: [Object],
    updatedAt: [Object]
  }
},
{
  Record: {
    productID: 'Paracetamol-001',
    name: 'Paracetamol',
    manufacturer: 'MAN001-Sun Pharma',
    manufacturingDate: '1/1/2020',
    expiryDate: '1/1/2023',
    owner: 'TRA001-FedEx',
    shipment: '',
    createdAt: [Object],
    updatedAt: [Object]
  }
},
{
  Record: {
    productID: 'Paracetamol-001',
    name: 'Paracetamol',
    manufacturer: 'MAN001-Sun Pharma',
    manufacturingDate: '1/1/2020',
    expiryDate: '1/1/2023',
    owner: 'MAN001-Sun Pharma',
    shipment: '',
    createdAt: [Object],
    updatedAt: [Object]
  }
}
```

- We can now also view the Drug Current State

Postman :

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (including 'View History' and 'View Drug Current State'), 'Environments', 'Most Servers', and 'Monitors'. The main area has a 'Scratch Pad' tab selected. A 'POST View Drug Current State' request is being edited. The URL is 'localhost:3000/viewDrugCurrentState'. The 'Body' tab is selected, showing a JSON payload:

```

{
  "drugname": "Paracetamol",
  "serialNo": "001",
  "organisationRole": "Consumer"
}

```

The response status is 200 OK, with a timestamp of 2.10 s and a size of 742 B. The response body is displayed in a code editor:

```

1
2   "status": "success",
3   "msg": "Drug current state found",
4   "output": [
5     {
6       "productID": "Paracetamol-001",
7       "name": "Paracetamol",
8       "manufacturer": "MAN001-Sun Pharma",
9       "manufacturingDate": "1/1/2020",
10      "expiryDate": "1/1/2023",
11      "owner": "AAD001",
12      "shipment": "\x00pharmanet.shipment\x00RET002-Upgrad-Paracetamol\x00",
13      "createdAt": {
14        "nanos": 493000000,
15        "seconds": {
16          "low": 1691580155,
17          "high": 0,
18          "unsigned": true
19        }
20      },
21      "updatedAt": {
22        "nanos": 493000000,
23        "seconds": {
24          "low": 1691580155,
25          "high": 0,
26          "unsigned": true
27        }
28      }
29    }
]

```

Console

```
{
  productID: 'Paracetamol-001',
  name: 'Paracetamol',
  manufacturer: 'MAN001-Sun Pharma',
  manufacturingDate: '1/1/2020',
  expiryDate: '1/1/2023',
  owner: 'AAD001',
  shipment: '\x00pharmanet.shipment\x00RET002-Upgrad-Paracetamol\x00',
  createdAt: {
    nanos: 493000000,
    seconds: { low: 1691580155, high: 0, unsigned: true }
  },
  updatedAt: {
    nanos: 493000000,
    seconds: { low: 1691580155, high: 0, unsigned: true }
  }
}
```

7. Clearing the setup files to re-launch the network

To clear the system with all the docker and other crypto materials we can run the following shell script

{Path_to_Project}/pharma-network/test-network-man/clearSetup.sh.

This script will only delete all the files that were created by script {Path_to_Project}/pharma-network/test-network-man/startNetwork.sh

It is recommended that before you run this command, you need to close the client side rest api application.