

# PHARMANET

Supply Chain Network for Drugs

## Fabric Network Setup

- The below command generates the crypto materials & channel artifacts. It also creates the docker containers, the channel and updates the anchor peers.

“./fabricNetwork.sh up”

```
+ set +x
2022-10-26 14:54:52.643 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-10-26 14:54:52.674 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
===== Anchor peers updated for org 'consumerMSP' on channel 'pharmachannel' =====

Updating anchor peers for MANUFACTURER...
+ peer channel update -o orderer.pharma-network.com:7050 -c pharmachannel -f ./channel-artifacts/manufacturerMSPanchors.tx
+ res=0
+ set +x
2022-10-26 14:54:57.825 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-10-26 14:54:57.885 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
===== Anchor peers updated for org 'ManufacturerMSP' on channel 'pharmachannel' =====

Updating anchor peers for DISTRIBUTOR...
+ peer channel update -o orderer.pharma-network.com:7050 -c pharmachannel -f ./channel-artifacts/distributorMSPanchors.tx
+ res=0
+ set +x
2022-10-26 14:55:03.233 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-10-26 14:55:03.260 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
===== Anchor peers updated for org 'distributorMSP' on channel 'pharmachannel' =====

Updating anchor peers for RETAILER...
+ peer channel update -o orderer.pharma-network.com:7050 -c pharmachannel -f ./channel-artifacts/retailerMSPanchors.tx
+ res=0
+ set +x
2022-10-26 14:55:08.461 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-10-26 14:55:08.516 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
===== Anchor peers updated for org 'retailerMSP' 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
2022-10-26 14:55:13.756 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-10-26 14:55:13.781 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! =====

[END]
upgrad@upgrad-VirtualBox:~/Desktop/PharmaNet_Capstone/network$
```

- Before proceeding to chaincode installation, make sure that the node modules are installed in the chaincode folder. To do so, follow the below commands

```
“cd ../chaincode”
“npm install”
“cd ../network”
```

```
+-- cross-spawn@5.1.0
+-- shebang-command@1.2.0
+-- shebang-regex@1.0.0
+-- get-stream@3.0.0
+-- is-stream@1.1.0
+-- npm-run-path@2.0.2
+-- path-key@2.0.1
+-- p-finally@1.0.0
+-- strip-eof@1.0.0
+-- lcld@1.0.0
+-- invert-kv@1.0.0
+-- mimic-fn@1.2.0
+-- require-directory@2.1.1
+-- set-blocking@2.0.0
+-- string-width@2.1.1
+-- is-fullwidth-code-point@2.0.0
+-- strip-ansi@4.0.0
+-- ansi-regex@3.0.0
+-- which-module@2.0.0
+-- y18n@3.2.1
+-- yargs-parser@9.0.2
+-- camelcase@4.1.0
+-- camelcase@4.1.0
+-- sinon@4.3.5
+-- @sinonjs/commons@1.8.3
+-- @sinonjs/formatio@3.2.2
+-- @sinonjs/samsam@3.3.3
+-- array-from@2.1.1
+-- @sinonjs/samsam@2.1.3
+-- lodash.get@4.4.2
+-- lodash@2.7.5
+-- ntlex@1.5.3
+-- @sinonjs/text-encoding@0.7.2
+-- just-extend@4.2.1
+-- lodash@5.1.2
+-- path-to-regexp@1.8.0
+-- tsarray@0.6.1
+-- supports-color@5.5.0
+-- sinon-chai@3.7.0
+-- which@1.3.1
+-- lsexec@2.0.0

npm WARN pharmanet@0.0.1 No repository field.
npm WARN pharmanet@0.0.1 No license field.
upgrad@upgrad-VirtualBox:~/Desktop/PharmaNet_Capstone/chaincodes$
```

- Now use the below commands to install and instantiate the chaincode on the peers

“./fabricNetwork.sh install”

```

Installing chaincode on peer0.retailer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.3 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chaincode/
+ res=0
+ set +x
2022-10-26 15:26:03.541 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escv
2022-10-26 15:26:03.541 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2022-10-26 15:26:03.685 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer0.retailer =====

Installing chaincode on peer0.transporter.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.3 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chaincode/
+ res=0
+ set +x
2022-10-26 15:26:03.839 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escv
2022-10-26 15:26:03.839 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2022-10-26 15:26:04.016 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer0.transporter =====

Installing chaincode on peer0.consumer.pharma-network.com ...
+ peer chaincode install -n pharmanet -v 1.3 -l node -p /opt/gopath/src/github.com/hyperledger/fabric/peer/chaincode/
+ res=0
+ set +x
2022-10-26 15:26:04.146 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escv
2022-10-26 15:26:04.146 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2022-10-26 15:26:04.358 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
===== Chaincode is installed on peer0.consumer =====

Instantiating chaincode on channel using peer0.manufacturer.pharma-network.com ...
+ peer chaincode instantiate -o orderer.pharma-network.com:7050 -C pharmanet -n pharmanet -l node -v 1.3 -c '{"Args":["org.pharma-network.pharmanet:instantiate"]}' -P 'OR (''manufacturerMSP.member'\'
'', ''distributorMSP.member'\'', ''retailerMSP.member'\'', ''transporterMSP.member'\'', ''consumerMSP.member'\'')'
+ res=0
+ set +x
2022-10-26 15:26:04.472 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escv
2022-10-26 15:26:04.472 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
===== Chaincode is installed on peer0.manufacturer on channel 'pharmanet' =====

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

END
upgrad@upgrad-VirtualBox:~/Desktop/PharmaNet_Capstone/network$

```

## Node Application Setup

- Navigate to the Application folder  
"cd ../application"
- Start the Node application by the following command  
"node ."
- \*If met with an error please check node version.

```

upgrad@upgrad-VirtualBox:~/Desktop/PharmaNet_Capstone/application$ node .
Distributed Pharma App listening on port 4000!
Distributor identity added to wallet.
Consumer identity added to wallet.
Transporter identity added to wallet.
Manufacturer identity added to wallet.
Retailer identity added to wallet.
Inside scope "DIST001"

```



## Postman Setup

- Open the Postman Application
- Click on the + icon to import collections
- Navigate to the test folder and import the whole folder to the Postman

## Testcase 1 - Initiation

- Run the collection in order to create companies with the given details
- Manufacturer Sun Pharma

The screenshot shows a REST client interface with a POST request to `localhost:4000/registerCompany`. The request body is a JSON object with the following details:

```
{  "nameOfOrg": "manufacturer",  "companyCRN": "MAN001",  "companyName": "Sun Pharma",  "location": "Chennai",  "organisationRole": "Manufacturer"}
```

The response is a 200 OK status with a response time of 3.44 s and a body size of 443 B. The response body is a JSON object:

```
{  "status": "success",  "message": "New Company got registered",  "newCompany": {    "companyID": "MAN001-Sun Pharma",    "name": "Sun Pharma",    "location": "Chennai",    "organisationRole": "Manufacturer",    "hierarchyKey": 1  }}
```

- Transporter “FedEx”

The screenshot shows a REST client interface with a POST request to `localhost:4000/registerCompany...`. The request body is a JSON object with the following details:

```
{  "nameOfOrg": "transporter",  "companyCRN": "TRA001",  "companyName": "FedEx",  "location": "Delhi",  "organisationRole": "Transporter"}
```

The response is a 200 OK status with a response time of 1 m 22.56 s and a body size of 430 B. The response body is a JSON object:

```
{  "status": "success",  "message": "New Company got registered",  "newCompany": {    "companyID": "TRA001-FedEx",    "name": "FedEx",    "location": "Delhi",    "organisationRole": "Transporter",    "hierarchyKey": 0  }}
```

- Distributor “VG Pharma”

The screenshot shows a REST client interface with a POST request to `localhost:4000/registerCompany`. The request body is a JSON object with the following details:

```
{  "nameOfOrg": "distributor",  "companyCRN": "DIST001",  "companyName": "VG Pharma",  "location": "Vizag",  "organisationRole": "distributor"}
```

The response is a 200 OK status with a response time of 3.73 s and a body size of 439 B. The response body is a JSON object:

```
{  "status": "success",  "message": "New Company got registered",  "newCompany": {    "companyID": "DIST001-VG Pharma",    "name": "VG Pharma",    "location": "Vizag",    "organisationRole": "distributor",    "hierarchyKey": 2  }}
```



- Transporter “Blue Dart”

POST localhost:4000/registerCompany...

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

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

```

1  {
2    "nameOfOrg": "transporter",
3    "companyCRN": "TRA002",
4    "companyName": "Blue Dart",
5    "location": "Bangalore",
6    "organisationRole": "Transporter"
7  }

```

Body Cookies Headers (7) Test Results 200 OK 2.65 s 442 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "status": "success",
3    "message": "New Company got registered",
4    "newCompany": {
5      "companyID": "TRA002-Blue Dart",
6      "name": "Blue Dart",
7      "location": "Bangalore",
8      "organisationRole": "Transporter",
9      "hierarchyKey": 0
10   }
11 }

```

- Retailer “upgrad”

POST localhost:4000/registerCompany...

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

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

```

1  {
2    "nameOfOrg": "retailer",
3    "companyCRN": "RET002",
4    "companyName": "upgrad",
5    "location": "Mumbai",
6    "organisationRole": "Retailer"
7  }

```

Body Cookies Headers (7) Test Results 200 OK 1 m 38.64 s 430 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "status": "success",
3    "message": "New Company got registered",
4    "newCompany": {
5      "companyID": "RET002-upgrad",
6      "name": "upgrad",
7      "location": "Mumbai",
8      "organisationRole": "Retailer",
9      "hierarchyKey": 3
10   }
11 }

```

- Run the balance files in order to add 4 strips of a drug named “Paracetamol” from “001 - 004”

POST localhost:4000/addDrug

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

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

```

1  {
2    "nameOfOrg": "manufacturer",
3    "drugName": "Paracetamol",
4    "serialNo": "001",
5    "mfgDate": "1-1-2020",
6    "expDate": "1-1-2025",
7    "companyCRN": "MAN001"
8  }

```

Body Cookies Headers (7) Test Results 200 OK 2.88 s 634 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "status": "success",
3    "message": "New Drug got registered",
4    "newDrug": {
5      "productID": "00org.pharma-network.pharmanet.drug00Paracetamol-0010",
6      "name": "Paracetamol",
7      "manufacturer": "00org.pharma-network.pharmanet.company0MAN0010",
8      "manufacturingDate": "1-1-2020",
9      "expiryDate": "1-1-2025",
10     "owner": "00org.pharma-network.pharmanet.company0MAN0010"
11   }
12 }

```

POST localhost:4000/addDrug Send

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

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

```
1 {
2   "nameOfOrg": "manufacturer",
3   "drugName": "Paracetamol",
4   "serialNo": "002",
5   "mfgDate": "2-2-2020",
6   "expDate": "2-2-2025",
7   "companyCRN": "MAN001"
8 }
```

Body Cookies Headers (7) Test Results 200 OK 2.42 s 634 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "New Drug got registered",
4   "newDrug": {
5     "productID": "0org.pharma-network.pharmanet.drug0Paracetamol-0020",
6     "name": "Paracetamol",
7     "manufacturer": "0org.pharma-network.pharmanet.company0MAN0010",
8     "manufacturingDate": "2-2-2020",
9     "expiryDate": "2-2-2025",
10    "owner": "0org.pharma-network.pharmanet.company0MAN0010"
11  }
12 }
```

POST localhost:4000/addDrug Send

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

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

```
1 {
2   "nameOfOrg": "manufacturer",
3   "drugName": "Paracetamol",
4   "serialNo": "003",
5   "mfgDate": "3-3-2020",
6   "expDate": "3-3-2025",
7   "companyCRN": "MAN001"
8 }
```

Body Cookies Headers (7) Test Results 200 OK 2.58 s 634 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "New Drug got registered",
4   "newDrug": {
5     "productID": "0org.pharma-network.pharmanet.drug0Paracetamol-0030",
6     "name": "Paracetamol",
7     "manufacturer": "0org.pharma-network.pharmanet.company0MAN0010",
8     "manufacturingDate": "3-3-2020",
9     "expiryDate": "3-3-2025",
10    "owner": "0org.pharma-network.pharmanet.company0MAN0010"
11  }
12 }
```

POST localhost:4000/addDrug Send

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

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

```
1 {
2   "nameOfOrg": "manufacturer",
3   "drugName": "Paracetamol",
4   "serialNo": "004",
5   "mfgDate": "4-4-2020",
6   "expDate": "4-4-2025",
7   "companyCRN": "MAN001"
8 }
9
```

Body Cookies Headers (7) Test Results 200 OK 2.87 s 634 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "New Drug got registered",
4   "newDrug": {
5     "productID": "0org.pharma-network.pharmanet.drug0Paracetamol-0040",
6     "name": "Paracetamol",
7     "manufacturer": "0org.pharma-network.pharmanet.company0MAN0010",
8     "manufacturingDate": "4-4-2020",
9     "expiryDate": "4-4-2025",
10    "owner": "0org.pharma-network.pharmanet.company0MAN0010"
11  }
12 }
```

## Testcase 2 - Supply Chain

### Supply Chain A

- Run the collection supply chain part a to perform the following functions.
- Create a Purchase Order raised by “VG Pharma” to purchase 3 strips of “Paracetamol(001,002,003)” from “Sun Pharma”

The screenshot shows a REST client interface with a POST request to `localhost:4000/createPO...`. The request body is a JSON object with the following fields:

```
{  "nameOfOrg": "distributor",  "buyerCRN": "DIST001",  "sellerCRN": "MAN001",  "drugName": "Paracetamol",  "quantity": "3"}
```

The response is a 200 OK status with a response time of 2.43 s and a body size of 582 B. The response body is a JSON object:

```
{  "status": "success",  "message": "New PO got registered",  "newPurchaseOrder": {    "poID": "0org.pharma-network.pharmanet.order0DIST001-Paracetamol0",    "name": "Paracetamol",    "quantity": "3",    "buyer": "0org.pharma-network.pharmanet.company0DIST0010",    "seller": "0org.pharma-network.pharmanet.company0MAN0010"  }}
```

- Now a Shipment object is created by “Sun Pharma” for the above generated Purchase Order, through “FedEx” as transporter.

The screenshot shows a REST client interface with a POST request to `localhost:4000/createShipment...`. The request body is a JSON object with the following fields:

```
{  "nameOfOrg": "manufacturer",  "buyerCRN": "DIST001",  "drugName": "Paracetamol",  "listOfAssets": "001,002,003",  "transporterCRN": "TRA001"}
```

The response is a 200 OK status with a response time of 2.60 s and a body size of 613 B. The response body is a JSON object:

```
{  "status": "success",  "message": "New Shipment got registered",  "newShipment": {    "shipmentID": "0org.pharma-network.pharmanet.shipment0DIST001-Paracetamol0",    "creator": "0org.pharma-network.pharmanet.company0MAN0010",    "assets": [      "001",      "002",      "003"    ]  }}
```

- The shipment is delivered to “VG Pharma” by “FedEx” .

The screenshot shows a REST client interface with a POST request to `localhost:4000/updateShipment...`. The request body is a JSON object with the following fields:

```
{  "nameOfOrg": "transporter",  "buyerCRN": "DIST001",  "drugName": "Paracetamol",  "transporterCRN": "TRA001"}
```

The response is a 200 OK status with a response time of 2.60 s and a body size of 627 B. The response body is a JSON object:

```
{  "status": "success",  "message": "New Updated Shipment got registered",  "newUpdatedShipment": {    "shipmentID": "0org.pharma-network.pharmanet.shipment0DIST001-Paracetamol0",    "creator": "0org.pharma-network.pharmanet.company0MAN0010",    "assets": [      "001",      "002",      "003"    ]  }}
```

## Supply Chain B

- Run the collection supply chain part a to perform the following functions.
- Create a Purchase Order raised by “upgrad” to purchase 2 strips of “Paracetamol(001,002)” from “VG Pharma”

POST localhost:4000/createPO... Send

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

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

```
1 {
2   "nameOfOrg": "retailer",
3   "buyerCRN": "RET002",
4   "sellerCRN": "DIST001",
5   "drugName": "Paracetamol",
6   "quantity": "2"
7 }
8
```

Body Cookies Headers (7) Test Results 200 OK 2.66 s 581 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "New PO got registered",
4   "newPurchaseOrder": {
5     "poID": "Org.pharma-network.pharmanet.orderRET002-Paracetamol",
6     "name": "Paracetamol",
7     "quantity": "2",
8     "buyer": "Org.pharma-network.pharmanet.companyRET002",
9     "seller": "Org.pharma-network.pharmanet.companyDIST001"
10  }
11 }
```

- Now a Shipment object is created by “VG Pharma” for the above generated Purchase Order, through “Blue Dart” as transporter.

POST localhost:4000/createShipment... Send

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

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

```
1 {
2   "nameOfOrg": "distributor",
3   "buyerCRN": "RET002",
4   "drugName": "Paracetamol",
5   "listOfAssets": "001,002",
6   "transporterCRN": "TRA002"
7 }
8
```

Body Cookies Headers (7) Test Results 200 OK 2.81 s 607 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "New Shipment got registered",
4   "newShipment": {
5     "shipmentID": "Org.pharma-network.pharmanet.shipmentRET002-Paracetamol",
6     "creator": "Org.pharma-network.pharmanet.companyDIST001",
7     "assets": [
8       "001",
9       "002"
10    ]
11  }
12 }
```

- The shipment is delivered to “upgrad” by “Blue Dart” .

POST localhost:4000/updateShipment... Send

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

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

```
1 {
2   "nameOfOrg": "transporter",
3   "buyerCRN": "RET002",
4   "drugName": "Paracetamol",
5   "transporterCRN": "TRA002"
6 }
7
```

Body Cookies Headers (7) Test Results 200 OK 2.57 s 621 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "New Updated Shipment got registered",
4   "newUpdatedShipment": {
5     "shipmentID": "Org.pharma-network.pharmanet.shipmentRET002-Paracetamol",
6     "creator": "Org.pharma-network.pharmanet.companyDIST001",
7     "assets": [
8       "001",
9       "002"
10    ]
11  }
12 }
```



## Supply Chain C

- Run the collection supply chain part a to perform the following functions.
- A Customer named “Akash” with Aadhar Number “AAD001” buys a strip of “Paracetamol (001)” from “upgrad” .

The screenshot shows a REST client interface with a POST request to `localhost:4000/retailDrug...`. The request body is a JSON object with the following fields:

```
{  "nameOfOrg": "retailer",  "drugName": "Paracetamol",  "serialNo": "001",  "retailerCRN": "RET002",  "customerAadhar": "AAD001"}
```

The response is a 200 OK status with a response time of 2.64 s and a body size of 734 B. The response body is a JSON object:

```
{  "status": "success",  "message": "New Purchase from customer",  "newPurchase": {    "productID": "0org.pharma-network.pharmanet.drug0Paracetamol-0010",    "name": "Paracetamol",    "manufacturer": "0org.pharma-network.pharmanet.company0MAN0010",    "manufacturingDate": "1-1-2020",    "expiryDate": "1-1-2025",    "owner": "AAD001"  }}
```

## Testcase 3 - History Tracking

- Run the collection in order to view current details & history of a drug.
- Customer “Akash” wants check the history of “Paracetamol(001)”

The screenshot shows a REST client interface with a POST request to `localhost:4000/viewHistory...`. The request body is a JSON object with the following fields:

```
{  "nameOfOrg": "consumer",  "drugName": "Paracetamol",  "serialNo": "001"}
```

The response is a 200 OK status with a response time of 1 m 52.32 s and a body size of 2.17 KB. The response body is a JSON object:

```
{  "status": "success",  "message": "historyOfDrug",  "historyOfDrug": [    {      "productID": "0org.pharma-network.pharmanet.drug0Paracetamol-0010",      "name": "Paracetamol",      "manufacturer": "0org.pharma-network.pharmanet.company0MAN0010",      "manufacturingDate": "1-1-2020",      "expiryDate": "1-1-2025"    }  ]}
```

- Customer “Akash” checks the current state of the drug.

The screenshot shows a REST client interface with a POST request to `localhost:4000/viewCurrentState...`. The request body is a JSON object with the following fields:

```
{  "nameOfOrg": "consumer",  "drugName": "Paracetamol",  "serialNo": "001"}
```

The response is a 200 OK status with a response time of 2.50 s and a body size of 721 B. The response body is a JSON object:

```
{  "status": "success",  "message": "currentState",  "currentState": {    "productID": "0org.pharma-network.pharmanet.drug0Paracetamol-0010",    "name": "Paracetamol",    "manufacturer": "0org.pharma-network.pharmanet.company0MAN0010",    "manufacturingDate": "1-1-2020",    "expiryDate": "1-1-2025",    "owner": "AAD001"  }}
```