There are 2 different parts for this project-

1. The go-client[Middle-layer] which interacts with the Hyperledger fabric and invokes the smart-contract which performs various operations on the blockchain (Hyperledger fabric). Basically what we are doing here is running a server in port 8080 and interacting with the blockchain through this server. So this server acts as both a client and a server (This is a common approach in the blockchain space).
2. The Hyperledger test-network setup : This is the blockchain where the data we are putting in is getting stored.

In the go-client we have 3 different api endpoints :

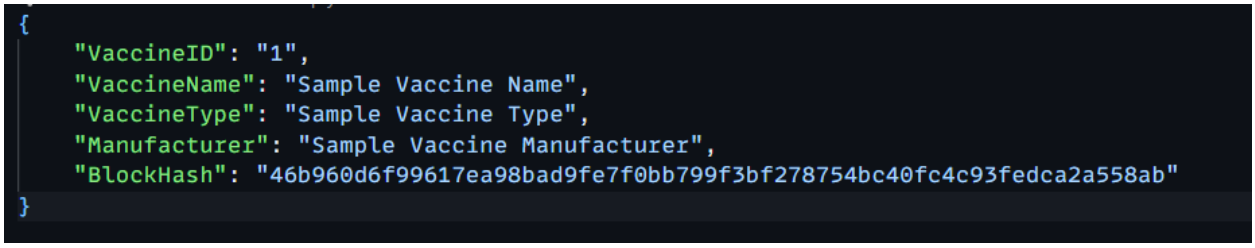
1. **[POST] /api/v1/add-vaccine-info**:
   This api is used to submit the vaccine details into the ledger or blockchain :

   Request : **localhost:8080/api/v1/add-vaccine-info**

   Body :

```
{
    "VaccineID": "3",
    "VaccineName": "Sample Vaccine Name 3",
    "AgeGroup": "1-20",
    "ContraIndications": "ContraIndications",
    "MethodOfAdministration": "MethodOfAdministration",
    "VaccineDescription": "VaccineDescription",
    "PharmaceuticalForm": "PharmaceuticalForm",
    "Precautions": "Precautions"
}
```

   Result :



2. **[POST] /api/v1/get-vaccine-details** :

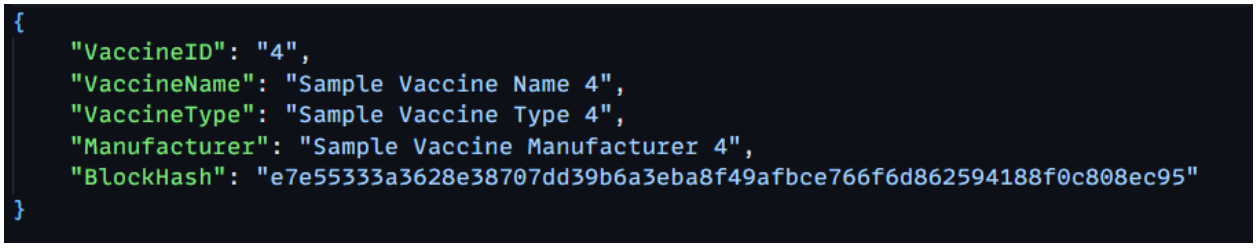
   This api is used to get the details regarding any vaccine which we have already stored into the ledger. The vaccine details or information is obtained using the case-id. Basically what this api does is like, it will query the ledger for the particular vaccine-id and will return the details or the information regarding that vaccine which is stored in the ledger.

Request : **localhost:8080/api/v1/get-vaccine-details**

Body :

```
{
"VaccineID":"1"
}
```

Result example :



3. **[GET] /api/v1/get-all-vaccine-details** :
   This request is used to get all the vaccine details which have been stored in the ledger.

   Request : **localhost:8080/api/v1/get-all-vaccine-details**
   Eg :

```
[
    {
        "VaccineID": "1",
        "VaccineName": "Sample Vaccine Name",
        "VaccineType": "Sample Vaccine Type",
        "Manufacturer": "Sample Vaccine Manufacturer",
        "BlockHash": "46b960d6f99617ea98bad9fe7f0bb799f3bf278754bc40fc4c93fedca2a558ab"
    },
    {
        "VaccineID": "2",
        "VaccineName": "Sample Vaccine Name 2",
        "VaccineType": "Sample Vaccine Type 2",
        "Manufacturer": "Sample Vaccine Manufacturer 2",
        "BlockHash": "5c8fbc31465808694a6596a858e2ecf7e0a3e93aff62c08e0d7a1f72027306c9"
    },
    {
        "VaccineID": "3",
        "VaccineName": "Sample Vaccine Name 3",
        "VaccineType": "Sample Vaccine Type 3",
        "Manufacturer": "Sample Vaccine Manufacturer 3",
        "BlockHash": "4963580842aea387fbda861377ef5f5fb79a06b602e6b623f2015f65dfa54c52"
    },
    {
        "VaccineID": "4",
        "VaccineName": "Sample Vaccine Name 4",
        "VaccineType": "Sample Vaccine Type 4",
        "Manufacturer": "Sample Vaccine Manufacturer 4",
        "BlockHash": "e7e55333a3628e38707dd39b6a3eba8f49afbce766f6d862594188f0c808ec95"
    }
]
```


4. **[POST]localhost:8080/api/v1/add-manufacture-detail**

   Body :

```
{
    "VaccineId":"3",
    "ManufacturerId":"0121"
}
```

5. **[POST]localhost:8080/api/v1/add-hospital-detail**

   Body :

```
{
    "VaccineId":"3",
    "HospitalId":"012"
}
```

6. **[POST]localhost:8080/api/v1/add-distributor-detail**

```
{
    "VaccineId":"3",
    "DistributorId":"0125"
}
```

7. **[POST]localhost:8080/api/v1/add-vaccine-expiry-detail**

```
{
    "VaccineId":"3",
    "ManufacturingDate":"2006-01-02",
    "ExpiryDate":"2008-01-02"
}
```

How to run the project ?

The root directory of the project is fabric-samples.
To run the hyperledger fabric node which we have configured, Hyperledger provides some scripts inside the directory test-network.
Listing the steps to be executed from the root directory :

1. **cd test-network** : This will change the current directory to test-network. The test-network directory contains all the dependency for the hyperledger-fabric network to run. This directory contains a network.sh script which helps us in running the hyperledger node inside docker and various other functions.
2. **./network.sh up** : This command will run the hyperledger instance inside docker.
3. Create a channel : In Hyperledger Fabric, a channel is a communication mechanism that allows different parties to interact privately and securely. It enables the segregation of transaction flows between different subsets of network participants. Channels provide confidentiality by restricting the visibility of transactions to only those parties that are part of the channel. This is a one-time thing, and needs to be done when we first set up the project.

   ./network.sh createChannel -c channel1
4. Deploy chaincode or smart contract to a channel : Deploying a chaincode to a channel in Hyperledger Fabric involves installing and instantiating the chaincode on the peers that are part of that channel. Here we are deploying the chaincode or smart contract written in go into channel1, which we have created above.
   ./network.sh deployCC -ccn vaccine1 -ccp ../fabric-setup/chaincode-go -ccl go -c channel1

5. Open another terminal in the root directory.
6. **cd fabric-setup/rest-api-go/** : This is the directory in which we have written the golang-client.
7. **go run main.go :** This command will run the golang-client which interacts with hyperledger fabric as well as the frontend.

Prerequisites Which needs to be installed to run the project :

1. Golang : go version go1.20.5 linux/amd64
2. Docker : Download docker from this link
https://desktop.docker.com/win/main/amd64/124339/Docker%20Desktop%20Installer.exe
[Version 4.24.2]

3. Use the command wsl -d Ubuntu-20.04 :
Inside wsl try these commands, this will install the required dependencies

1. sudo apt-get install git curl docker-compose -y

Make sure the Docker daemon is running.
2. sudo systemctl start docker

Add your user to the Docker group.
3. sudo usermod -a -G docker <username>

Check version numbers
4. docker --version
5. docker-compose --version

To deploy the new contract : [Should be done only one time]
./network.sh deployCC -ccn vaccine-contract -ccp ../fabric-setup/chaincode-go -ccl go -c channel1