Steps to understand the project Insurance Management app.

step-1 (Run app)

Please visit the class which is -> InsuranceManagementSystemApplication

Then right click on this and run the project

step-2

Jwt token authorization is enabled in this project, so without token generation,

this program will always give unauthorized 401 response.

Always pass the username as testABC and password as test123, to get the JWT token,

as in this project there is no need of authentication, so I haven't created a controller for user authentication.

Therefore, passing a static username and password.

Also proper validation and exception-handling on the data is also been implemented to avoid conflicts like runtime-errors.

In memory database h2 database is used in this project.

step-3

Its a gradle based spring boot server app project.

Dependencies are imported and downloaded via gradle build.

step-4

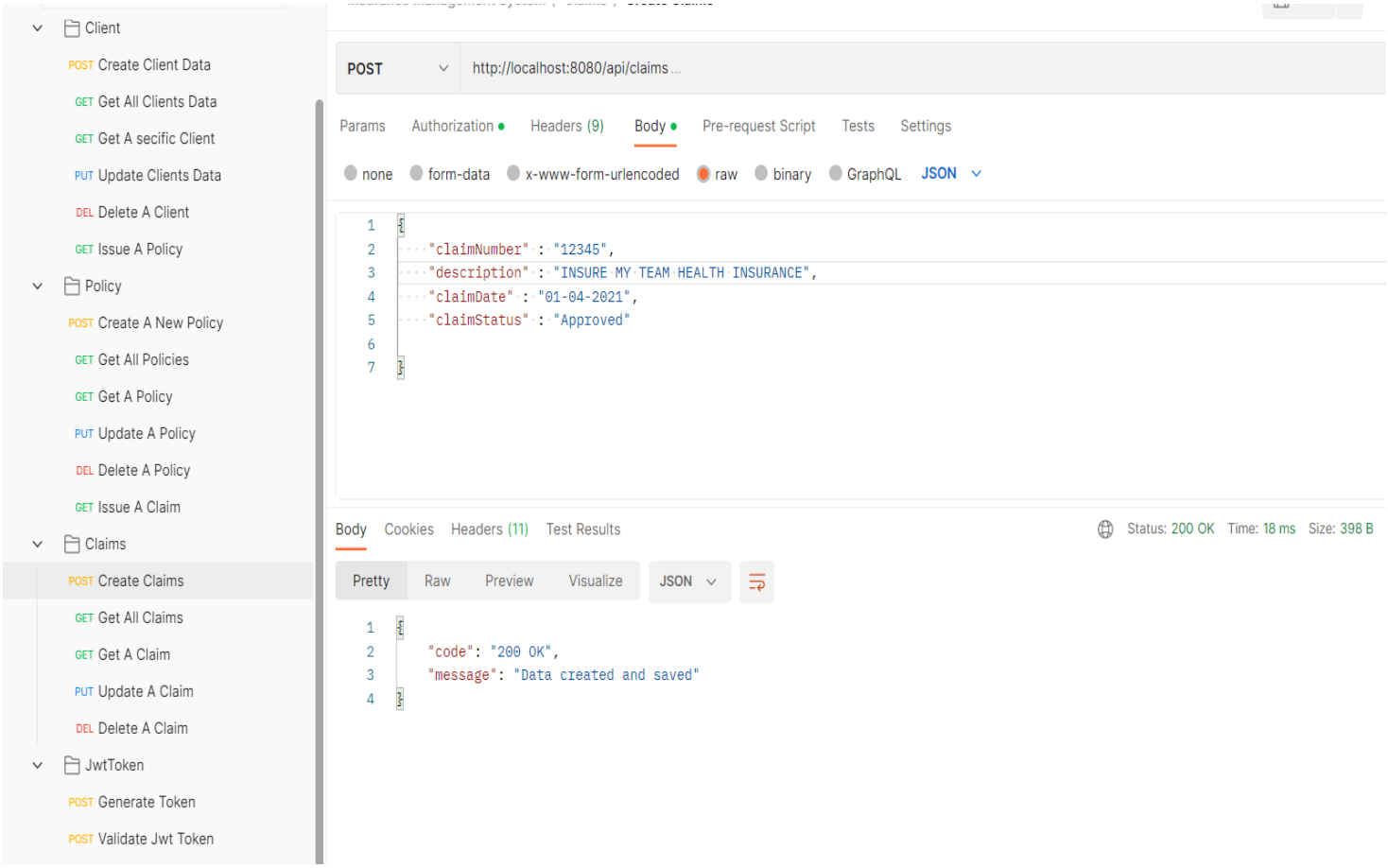
I am providing post man collection to test this project end-to-end,

with every features cleary mentioned and explained.

step-5

For Client, InsurancePolicy and Claims all the features are created

using the rest-api paradigm in spring boot.



step-6 (Structure of the project)

In every 5 minutes we have to validate jwt token, incase the token is expired then we should go for a new token generation api-call.

Entities created in this project are:

1) Client --> Client has 2 other sub-entities with one-to-one mapping wiht each sub-entities.

These are --> Address, ContactInformation

2) Insurance

3) Claim

Here, Client also has a one-to-one relation with insurance (insurance policy).

So, client can subscribe for a policy and get the benefits at the time of policy claims.

Also, Insurance (Insurance policy) has a one-to-one relation with the claims.

So, each policy has a claim, so that the client can avail the benefits of the claims whatever is associated with the policy.

API Processing steps

--> 1st client is created (POST mapping)

--> 2nd we can get client's data all at a time, or by id (GET mapping)

--> 3rd we can update client's data by passing their individual id (PUT mapping)

--> 4th we can delete client's data (DELETE mapping)

--> 5th we can issue policy to a client (PUT mapping)

similarly, we will do the same API mapping for Insurance policy and claims

Note- Policy is also linked with Insurance policy -> (Insurance policy has one-to-one mapping with claims).