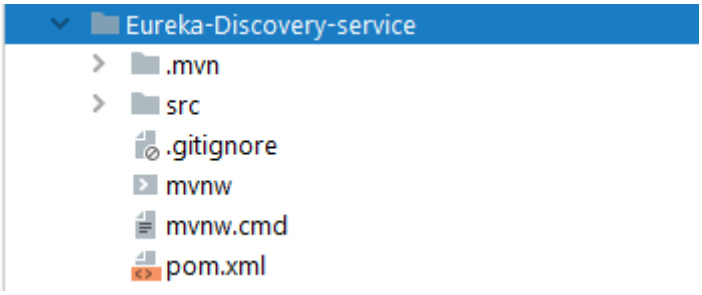
# Examen blanche Systèmes Distribués

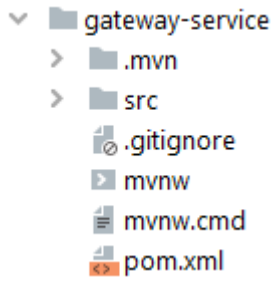
Réaliser par : Maarouf Ouassime

On souhaite créer un système distribué basé sur les micro-services en utilisant une architecture pilotée par les événements respectant les deux patterns Event Sourcing et CQRS. Cette application devrait permettre de gérer les infractions concernant des véhicules suites à des dépassement de vitesses détectés par des radars automatiques.

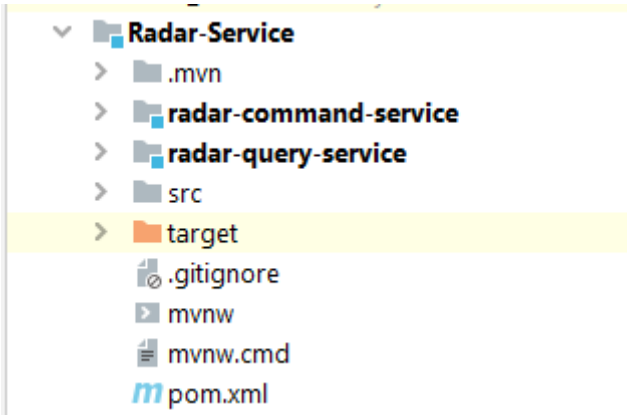
**Eureka discovery service :**



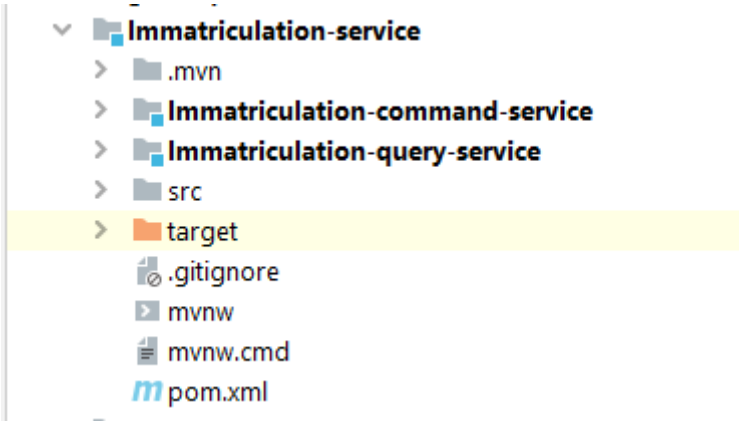

**gateway service :**



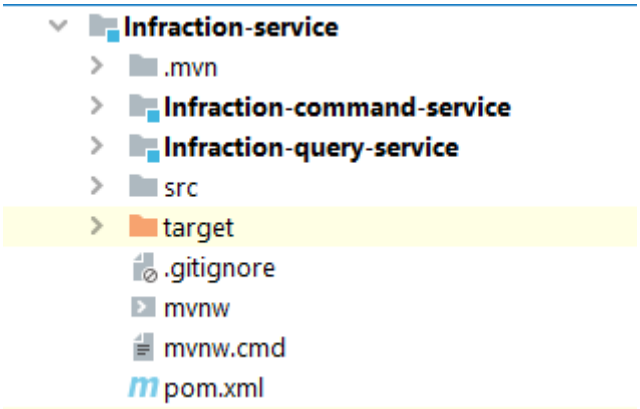

**radar service :**




**immatricualtion service :**

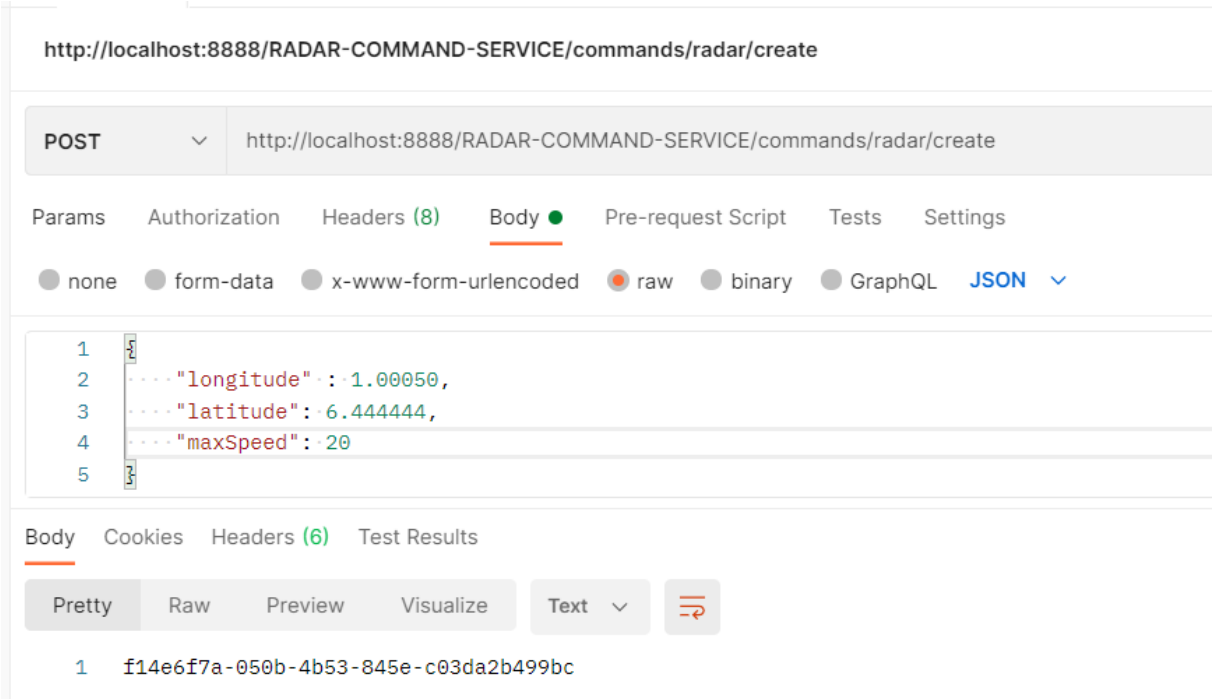



**infraction service :**

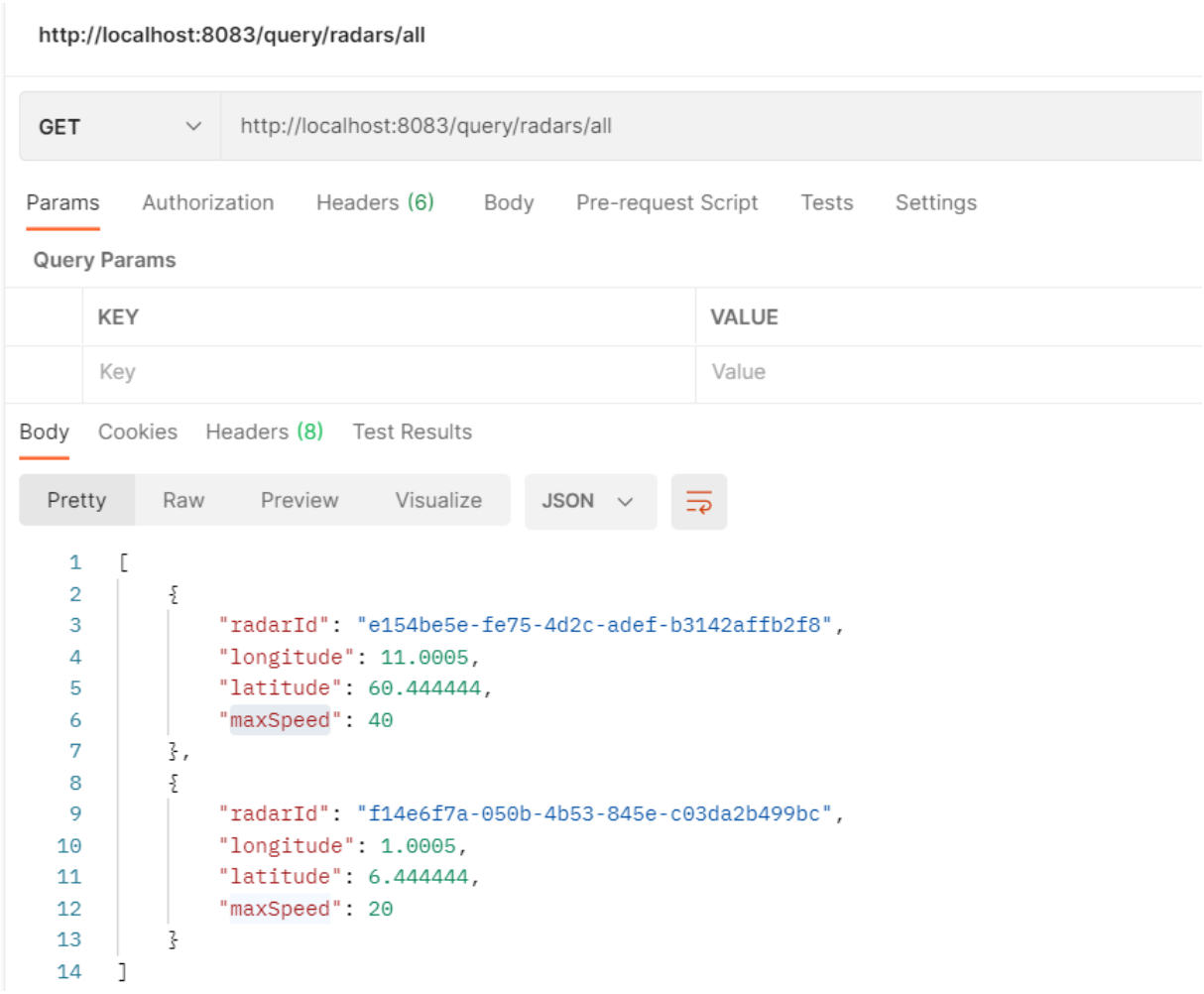



**Testing Apis :**

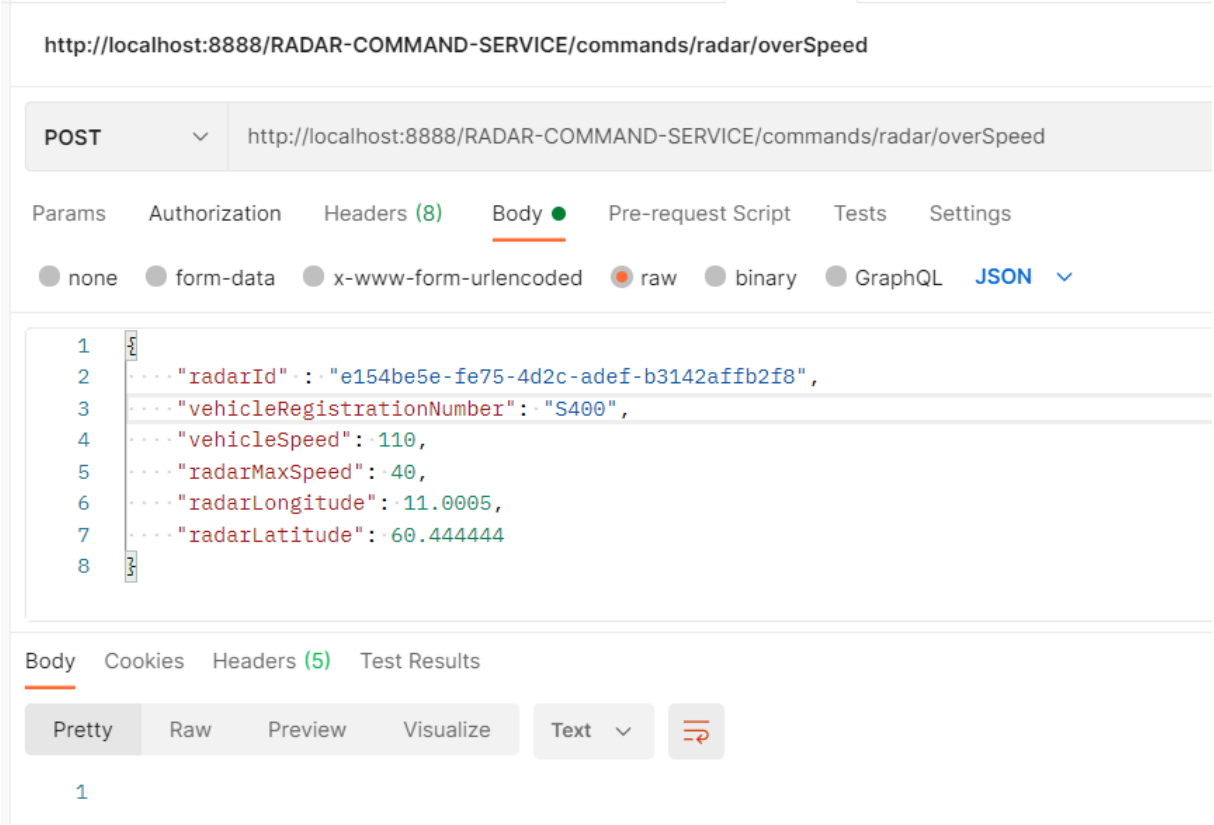
**ajouter radar :**



**get all radars :**




**over speed violation :**




**violation list :**

http://localhost:8888/RADAR-QUERY-SERVICE/query/overSpeeds/all

GET http://localhost:8888/RADAR-QUERY-SERVICE/query/overSpeeds/all

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

| KEY | VALUE |
| --- | --- |
| Key | Value |

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON


```
[
    {
        "overSpeedId": "1a23c2e8-2f5e-45c1-b746-366d981dd307",
        "timeStamp": "2022-12-24T17:08:21.554Z",
        "radarId": "",
        "vehicleRegistrationNumber": "S400",
        "vehicleSpeed": 110
    },
    {
        "overSpeedId": "512a3494-916e-477d-84dd-1d160815c275",
        "timeStamp": "2022-12-24T20:32:58.678Z",
        "radarId": "",
        "vehicleRegistrationNumber": "S400",
        "vehicleSpeed": 110
```

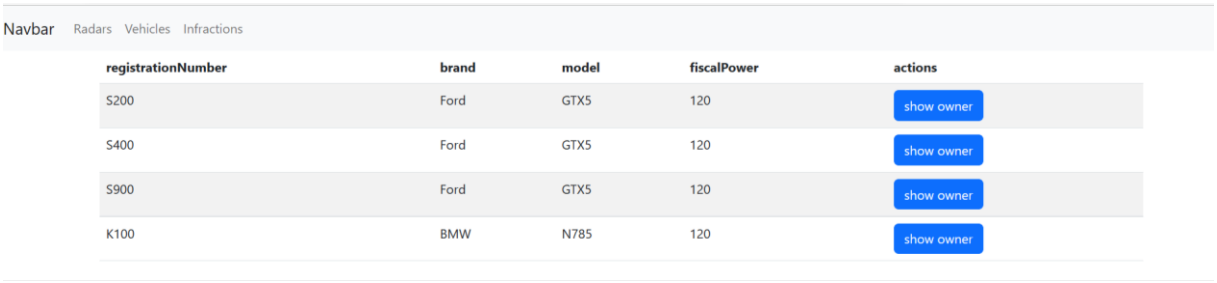

**Front With Angular Test :**

**radars list :**

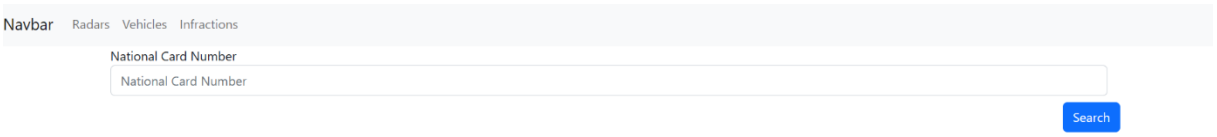
Navbar Radars Vehicles Infractions

| longitude | latitude | maxSpeed |
| --- | --- | --- |
| 11.0005 | 60.444444 | 40 |
| 1.0005 | 6.444444 | 20 |
| 22.0005 | 17.444444 | 60 |
| 72.0005 | 7.444444 | 50 |

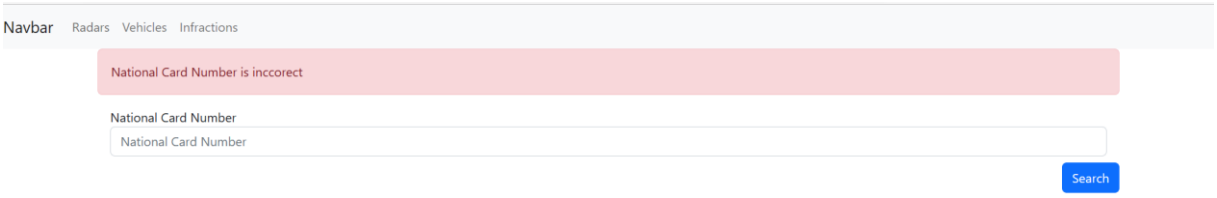
**vehicles list :**



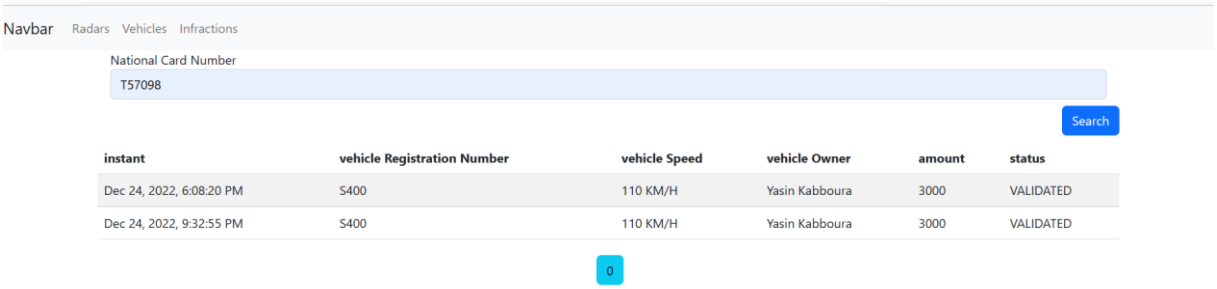

## search infractions




## infractions wrong National Card Number :

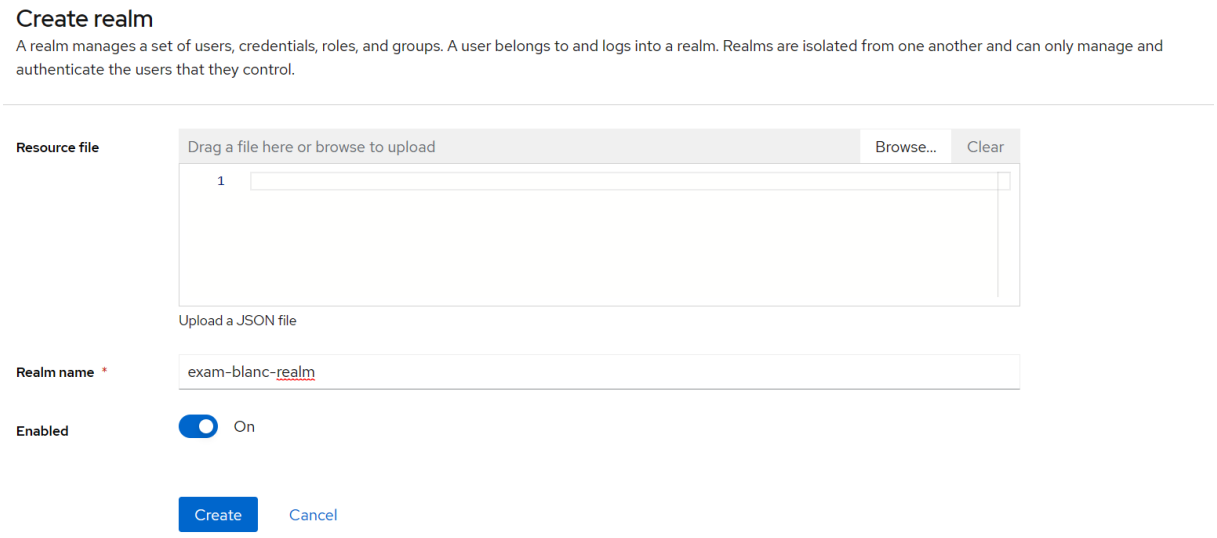



## infractions list :

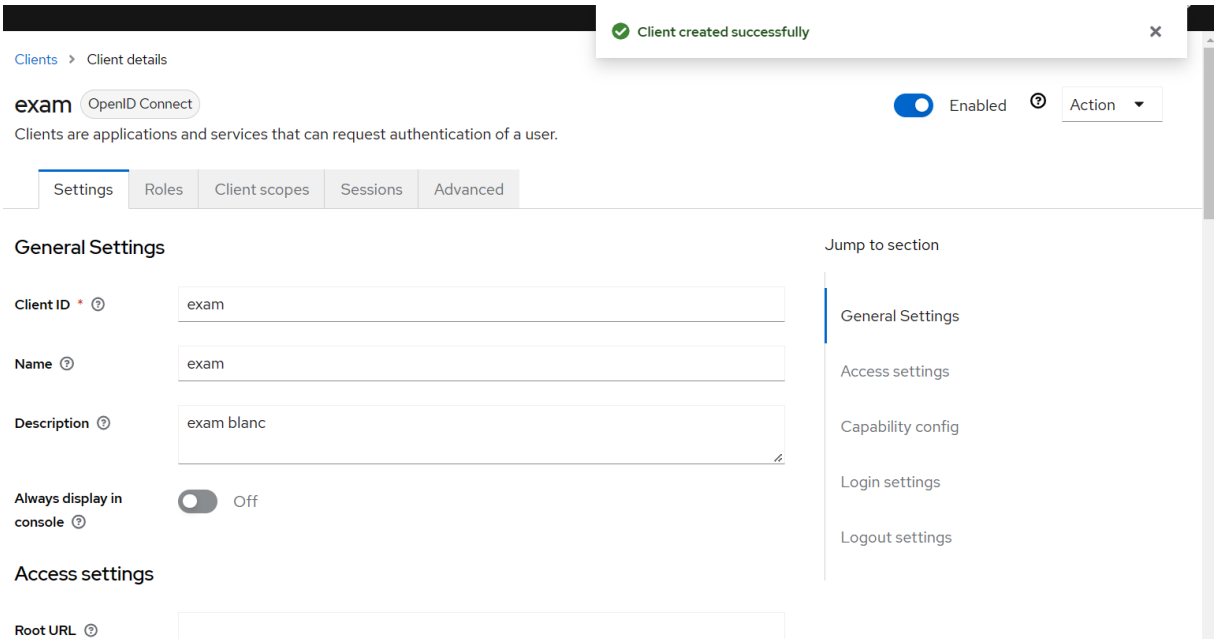



## Security with keycloack Test :

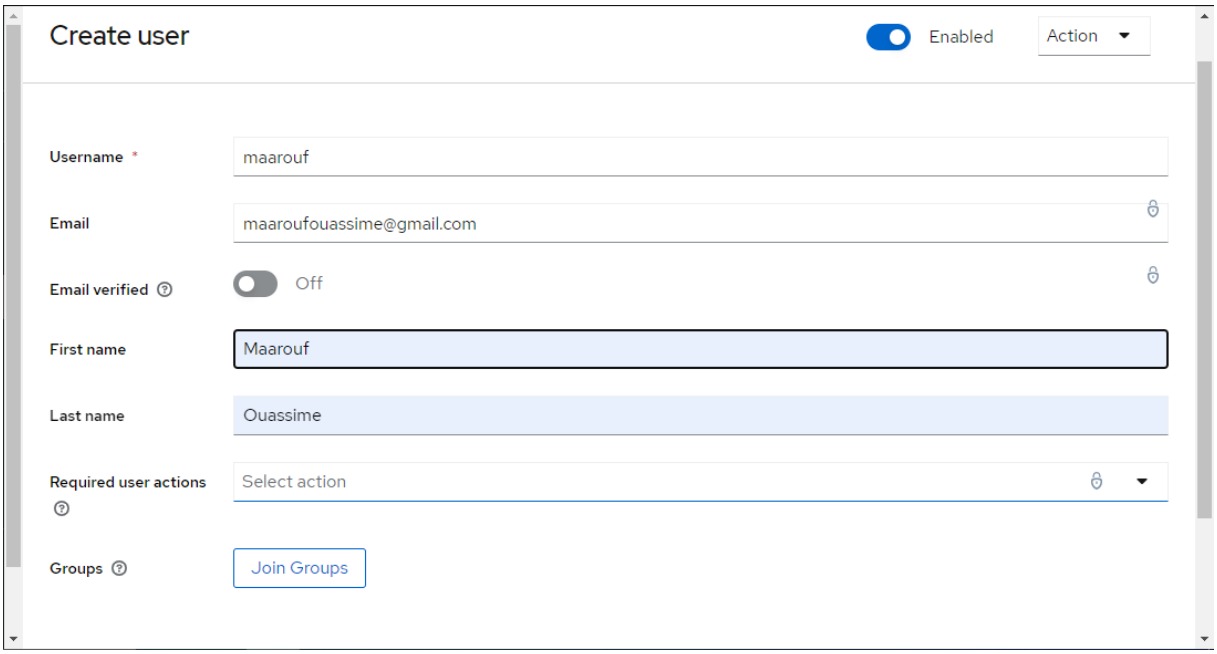
## creating exam-blanc -realm



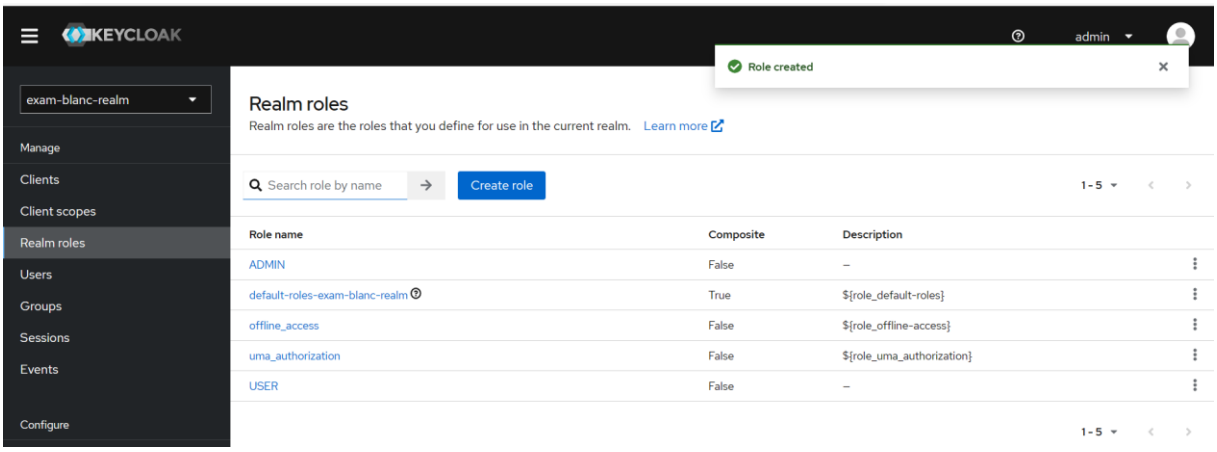

## creating exam client :



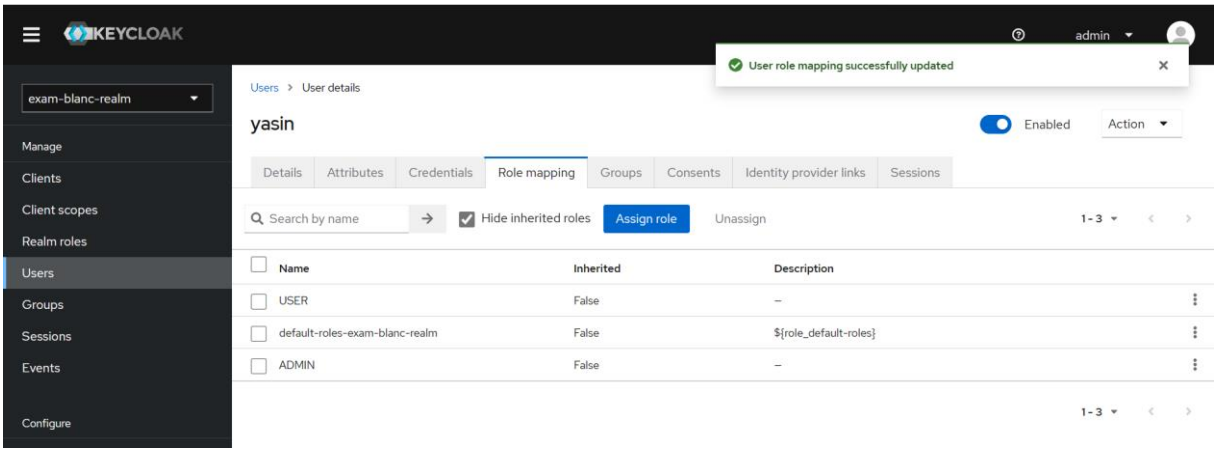

## creating users :



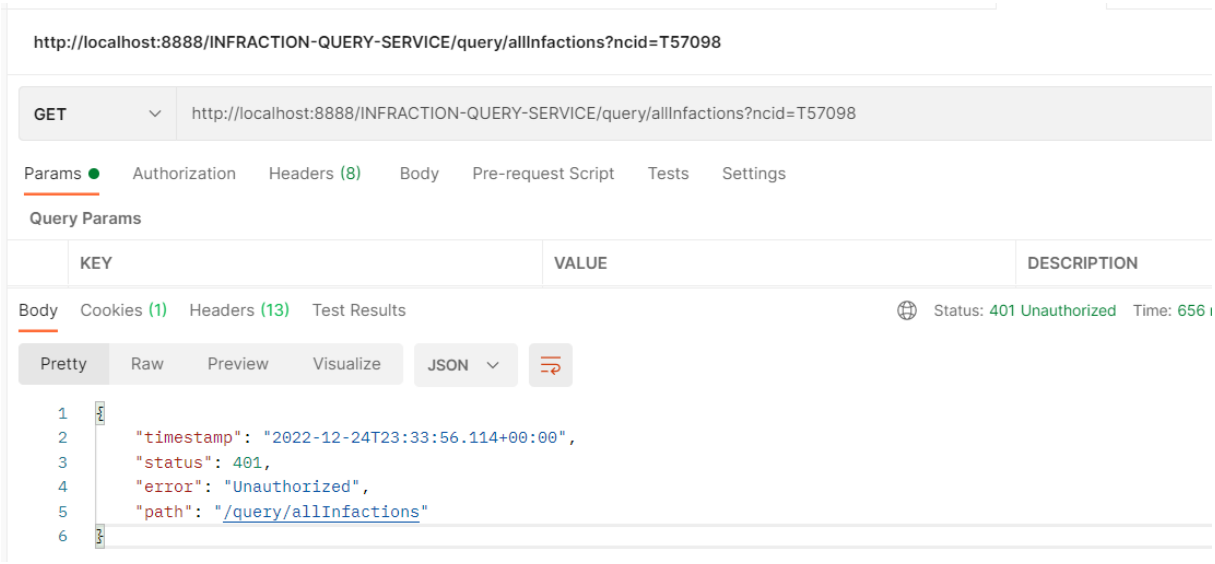

**creating roles ADMIN USER :**



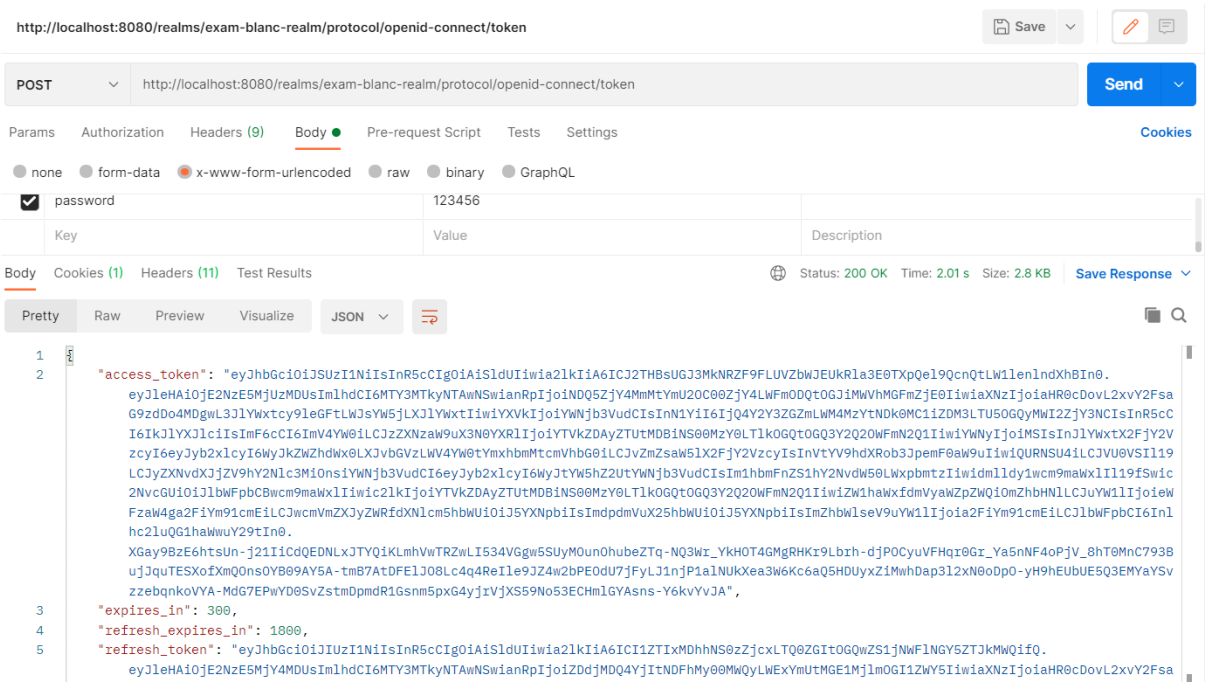

**givving the user roles :**



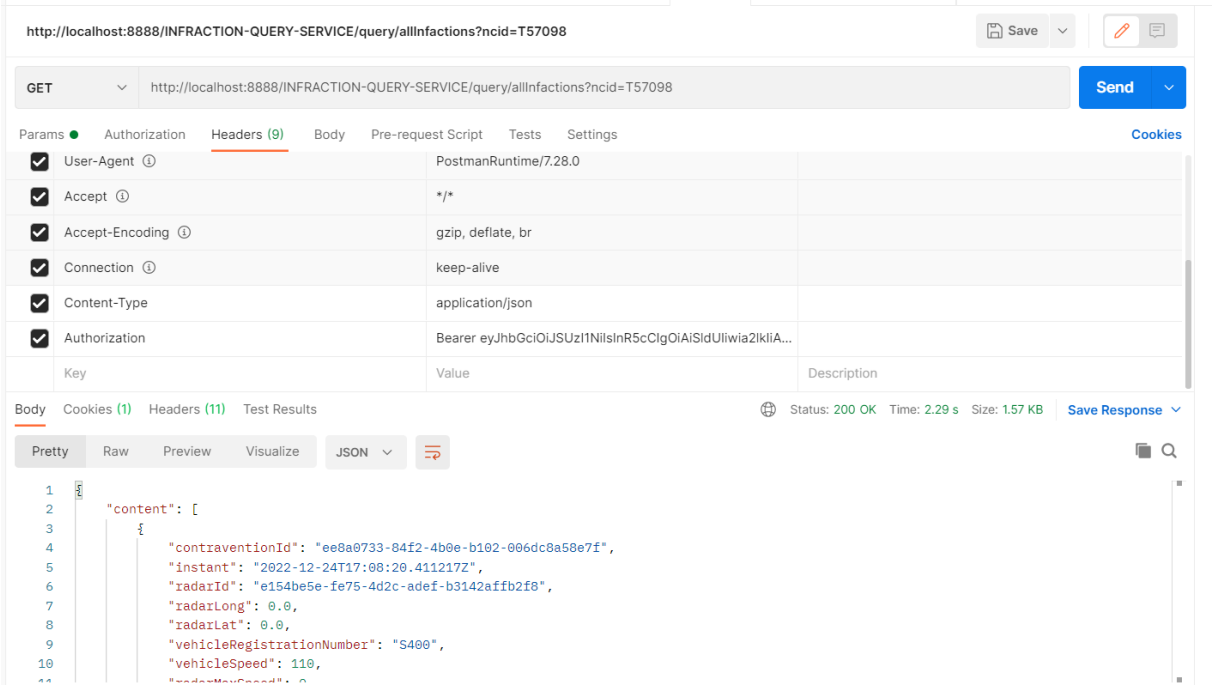

**test**




## generating jwt :




## Retesting :


http://localhost:8888/INFRACTION-QUERY-SERVICE/query/allInfactions?ncid=T57098
Save
GET
http://localhost:8888/INFRACTION-QUERY-SERVICE/query/allInfactions?ncid=T57098
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
User-Agent
PostmanRuntime/7.28.0
Accept
*/*
Accept-Encoding
gzip, deflate, br
Connection
keep-alive
Content-Type
application/json
Authorization
Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA...
Key
Value
Description
Body
Cookies (1)
Headers (11)
Test Results
Status: 200 OK
Time: 2.29 s
Size: 1.57 KB
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"content": [
{
"contraventionId": "ee8a0733-84f2-4b0e-b102-006dc8a58e7f",
"instant": "2022-12-24T17:08:20.411217Z",
"radarId": "e154be5e-fe75-4d2c-adef-b3142affb2f8",
"radarLong": 0.0,
"radarLat": 0.0,
"vehicleRegistrationNumber": "S400",
"vehicleSpeed": 110,