Spring Boot demo app:

Requirements:

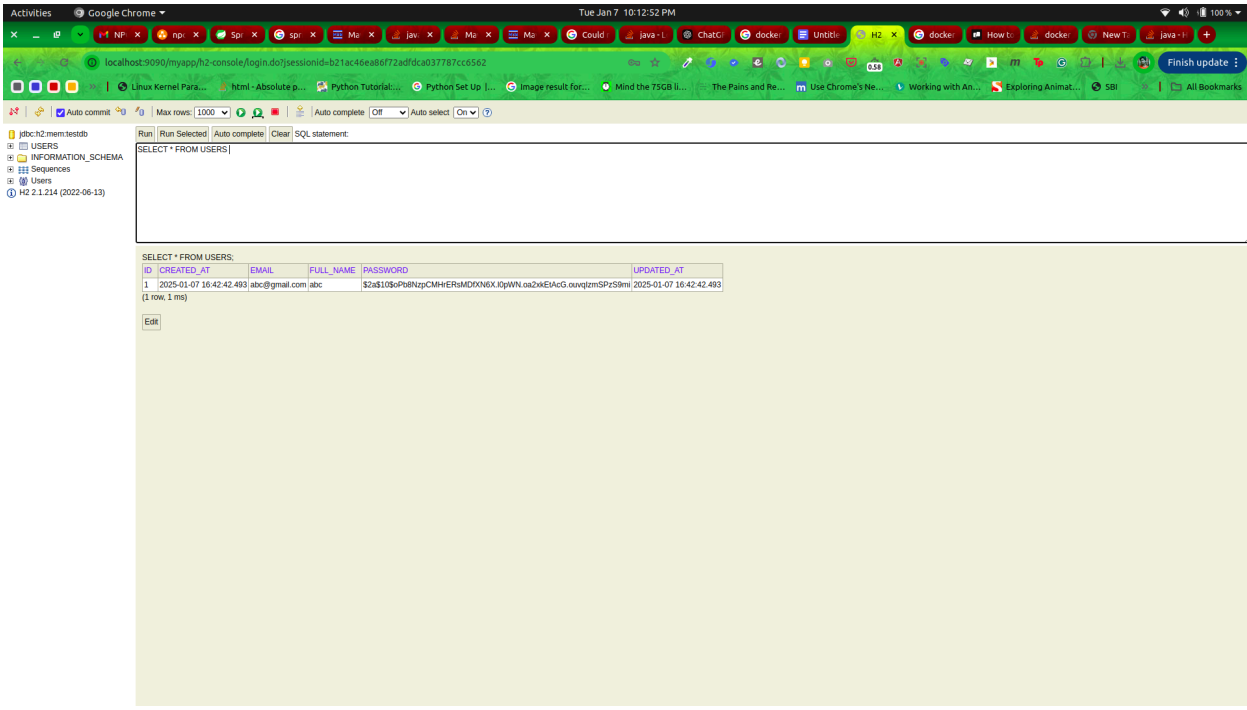
1. Java 8
2. Linux
3. Docker
4. Git (optional) - git commits are done

Functional Features:

1. JWT Authentication:
    a. Working context: Works based on signup user via signup apis
2. Rate Limiting
    a. Working context: based all over application per IP for X seconds of rate limited for Y number of request.
        i. X and Y configurable in /demo/src/main/java/com/example/demo/filter/RateLimitingFilter.java

Non-Functional Features:

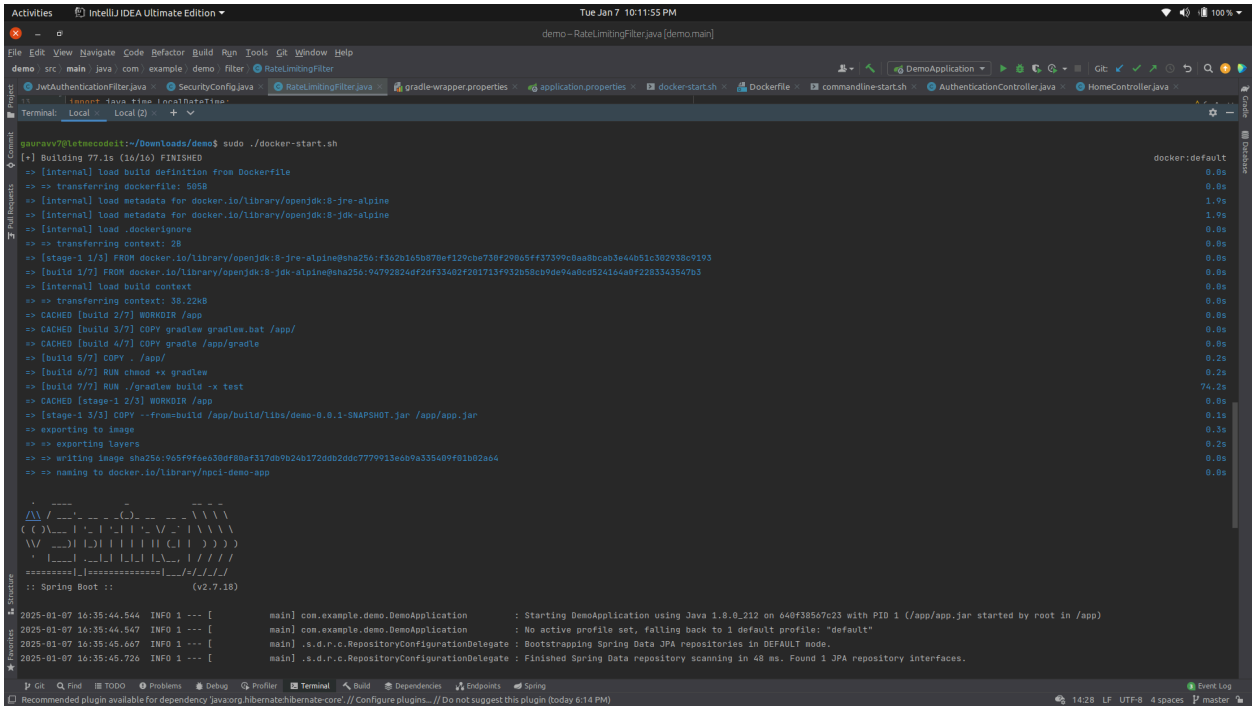
1. Spring boot based: version 2.7.18
    a. Gradle - v8
2. Embedded tomcat
    a. Runs on 9090
3. POSTMAN collection PFA in source code zip
4. Rest controllers - json apis
5. JWT auth based via user signup apis
6. H2 memory database used:
    a. Username: sa
    b. Password: sa
    c. Url: http://localhost:9090/myapp/h2-console/login.do
    d. Db url to be used: `jdbc:h2:mem:testdb`
    e. Screenshot

f. 

7. Rate limit configs available in code for
    a. MIN_IN_SECONDS_TO_RATE_LIMIT : defines the seconds until the rate limit is over.
    b. MAX_REQUESTS_PER_MINUTE defines the count threshold of requests when rate limit starts
8. Dockerization of app:
    a. The gradlew of the project will make the runnable jar in docker build stage and then docker will run the jar exposing the apis of the projects

SETUP:
run can be done in 2 ways

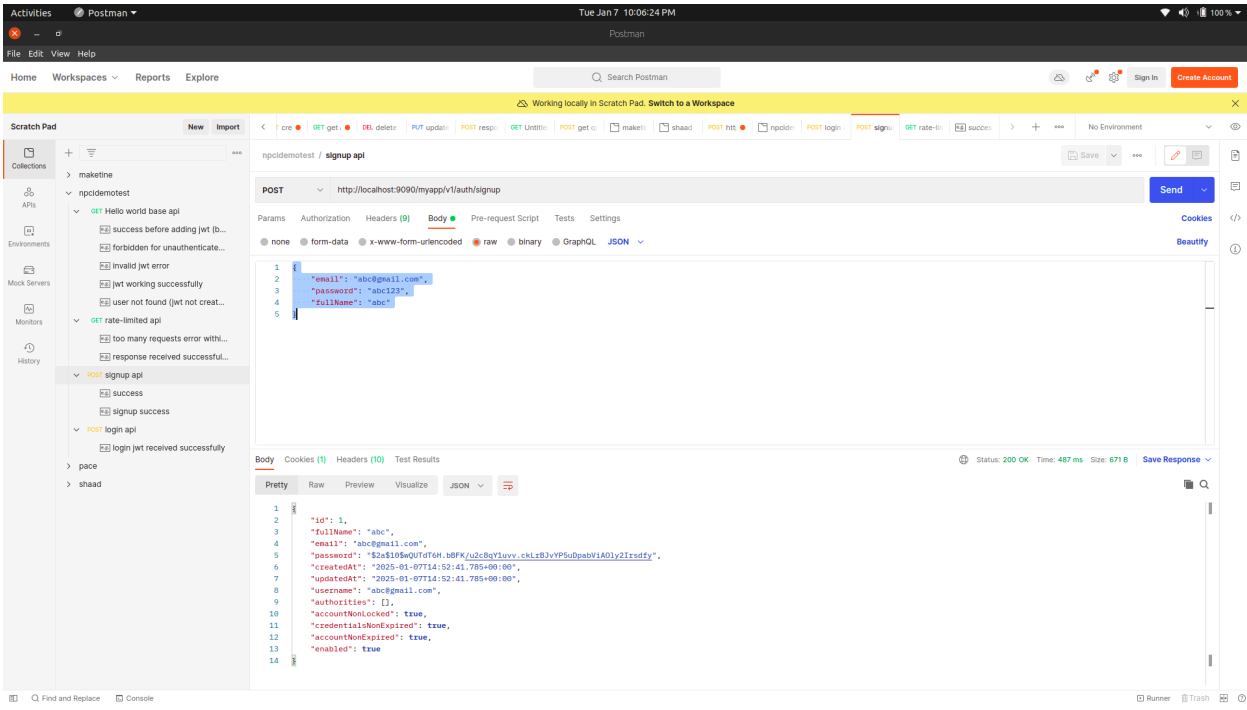
1. Via commanline-start.sh
    a. Command: sh commandline-start.sh
    b. Brief: this shall run the project on linux command without docker
2. Via docker-start.sh
    a. Command: sh docker-start.sh
    b. Brief: this shall run the project on linux command line with docker
    c. Requirement:
        i. Docker has to be installed

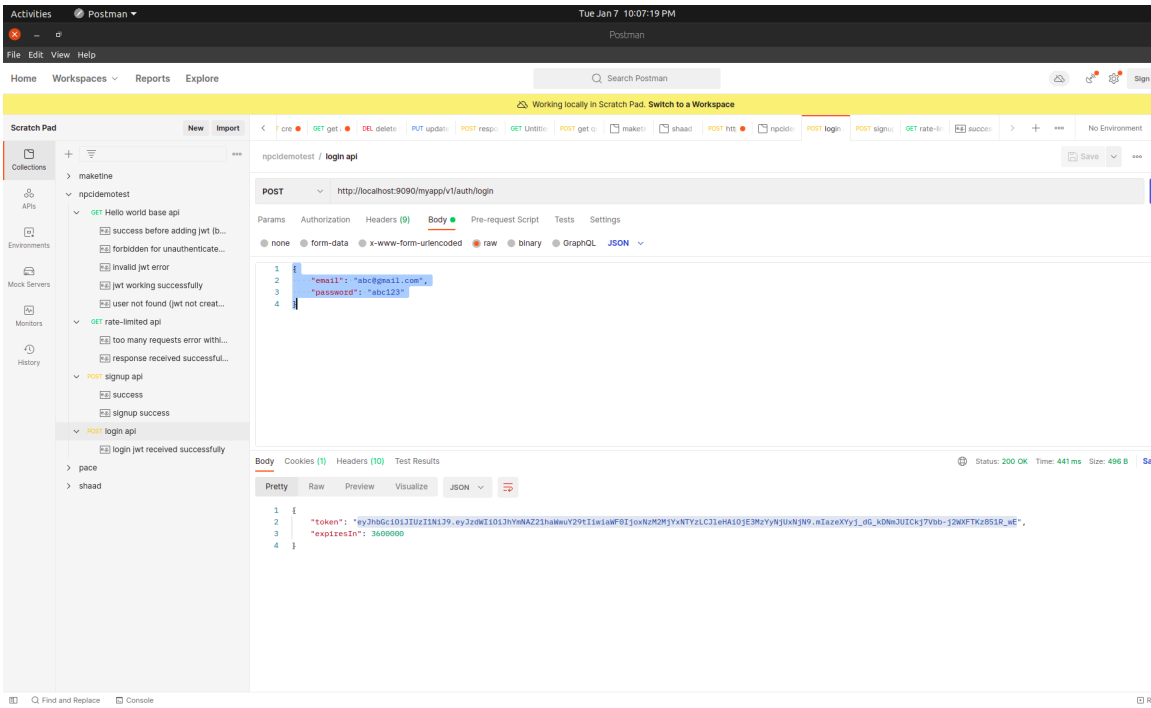


Project Port: 9090
Project api base path: /myapp

APIs:

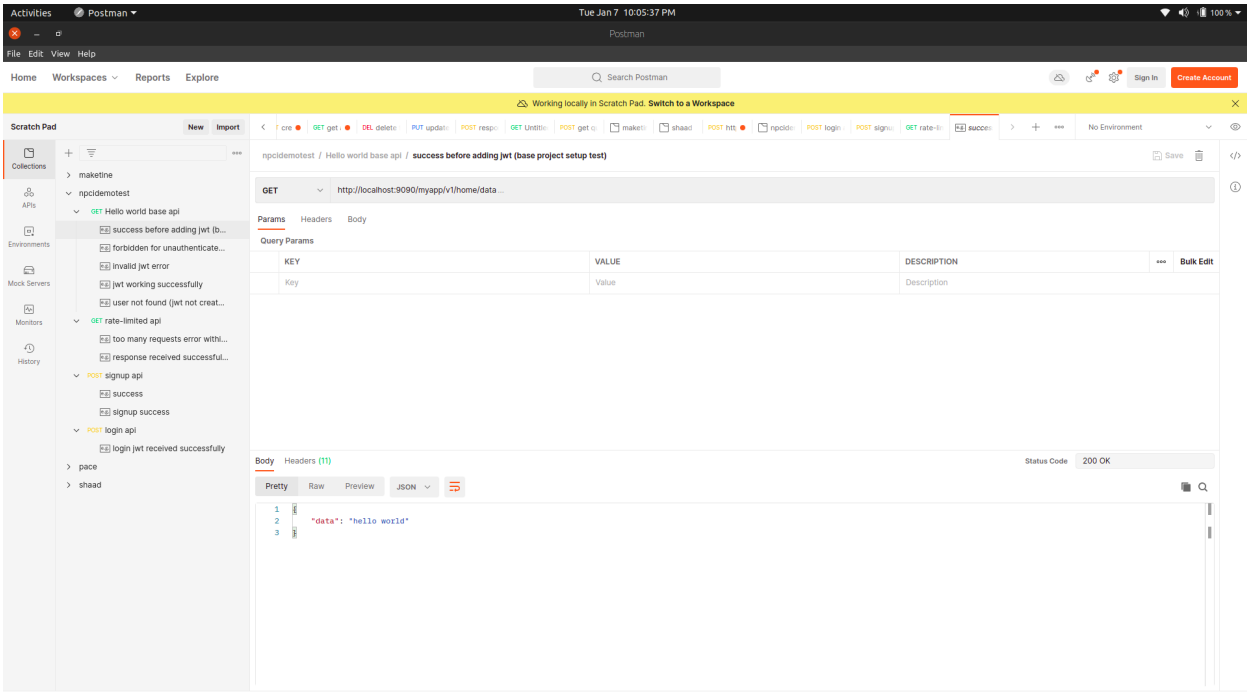
1. Signup API: to register a user for getting jwt after login:
    a. /myapp/v1/auth/signup HTTP POST
    b. Payload:
        i. {
        ii. "email": "abc@gmail.com",
        iii. "password": "abc123",
        iv. "fullName": "abc"
        v. }

c. 

2. Login up - to get the JWT token:
    a. /myapp/v1/auth/login HTTP POST
    b. Payload:
        i. {
        ii. "email": "abc@gmail.com",
        iii. "password": "abc123"
        iv. }
    v. 

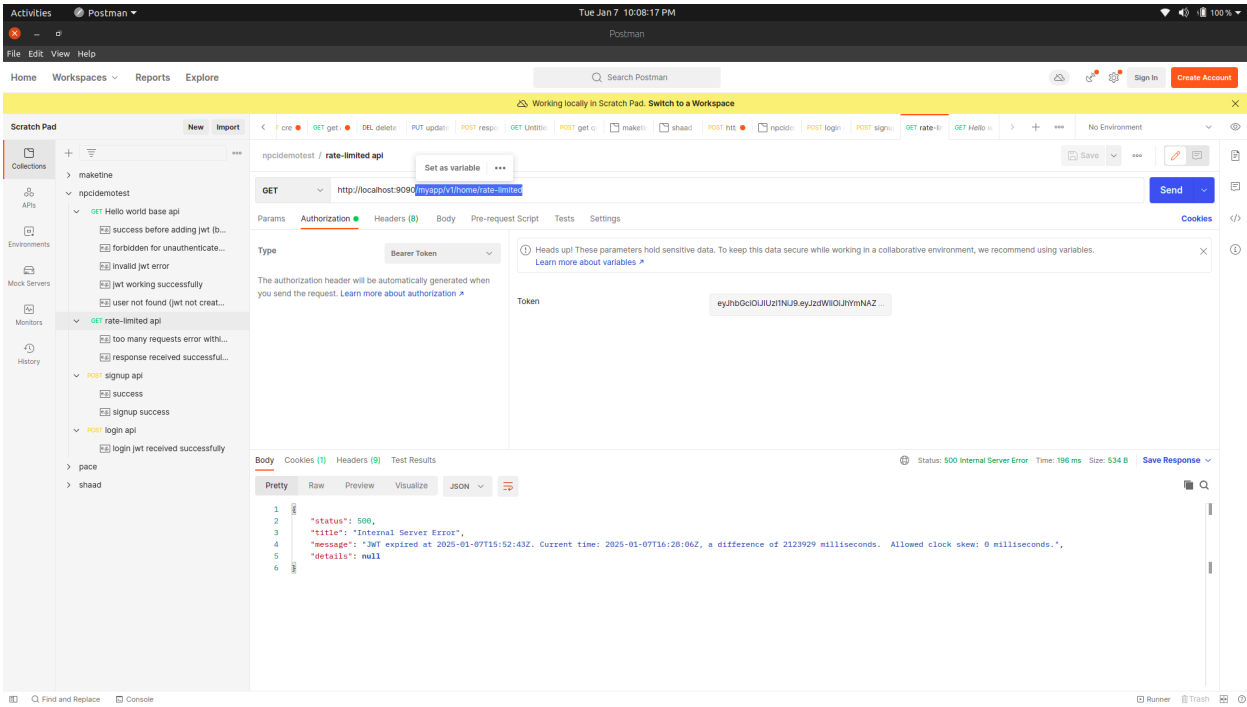
3. Data api - generic api: HTTP GET

a. /myapp/v1/home/data - working with JWT token



b.

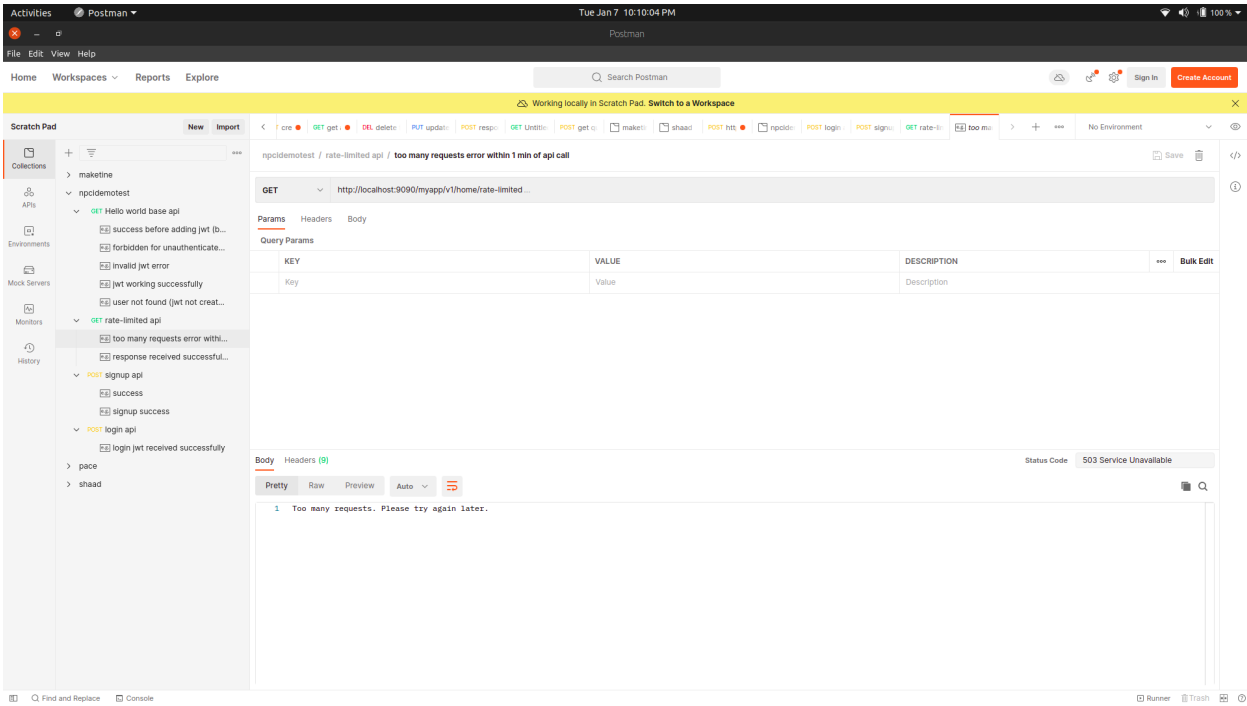
4. Data api - invalid JWT error
   a. HTTP GET
   b. /myapp/v1/home/rate-limited



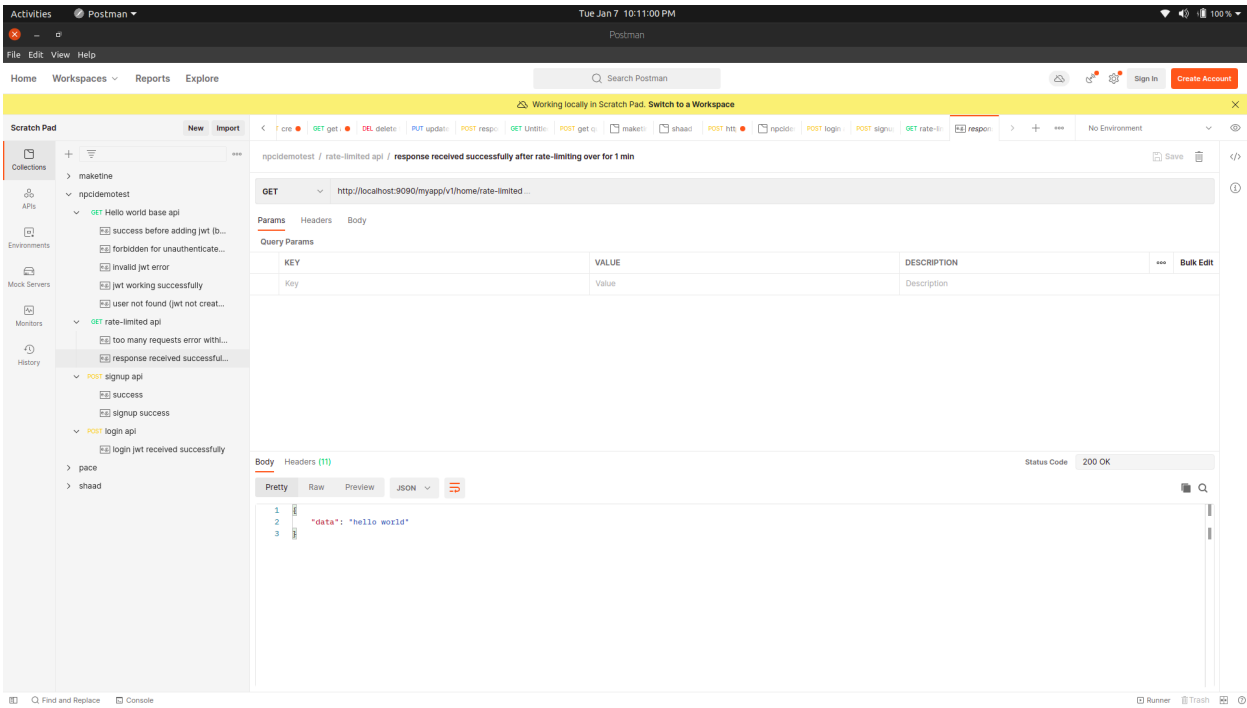
c.

5. Data api - generic HTTP GET with rate limited error
   a. HTTP GET
   b. /myapp/v1/home/rate-limited

c.



6. Data api - rate limited api but response received within rate limiting threshold
   a. /myapp/v1/home/rate-limited



b.