**Skill Tracker:**

MongoDB

Skilltracker-cmd- service

MySQL DB

Kafka

API Gateway

Client

OAUTH-JWT

Skill-tracker-query service

Skill-tracker-service discovery

Skill-tracker-config service

GIT

**Design Pattern:**

The application developed using mediator pattern. it is behavioral design pattern

Promotes loosely coupling by preventing objects from referring from each other explicitly

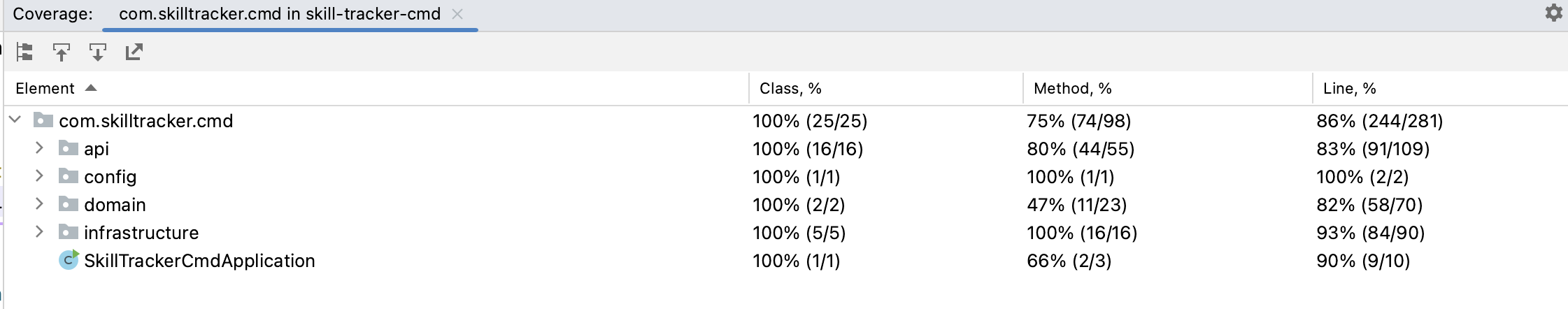
Simplifies communication between objects by introducing single object known as the mediator that manages the distribution of messages among other objects

**Number of Services developed/Used:**

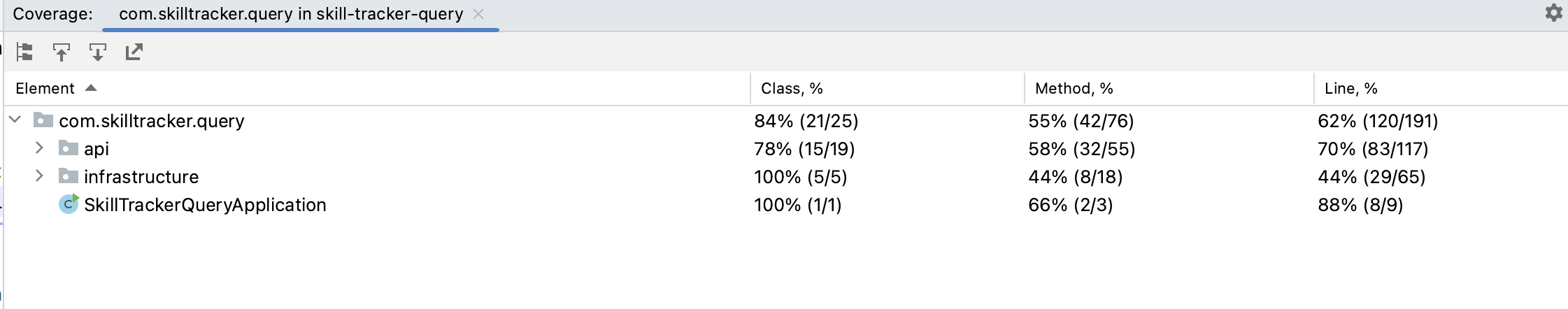
1. Register User
2. Create User
3. Update User
4. Restore DB
5. Create Profile
6. Update Profile
7. Search User
8. Search Profile
9. MongoDB
10. Kafka
11. MySQL DB
12. API Gateway
13. CircuitBreaker in API Gateway
14. Service Discovery
15. Spring Cloud config server
16. OAUTH

**Test coverage:**

Skill-Tracker-cmd



Skill-Tracker-Query



**Screenshots:**

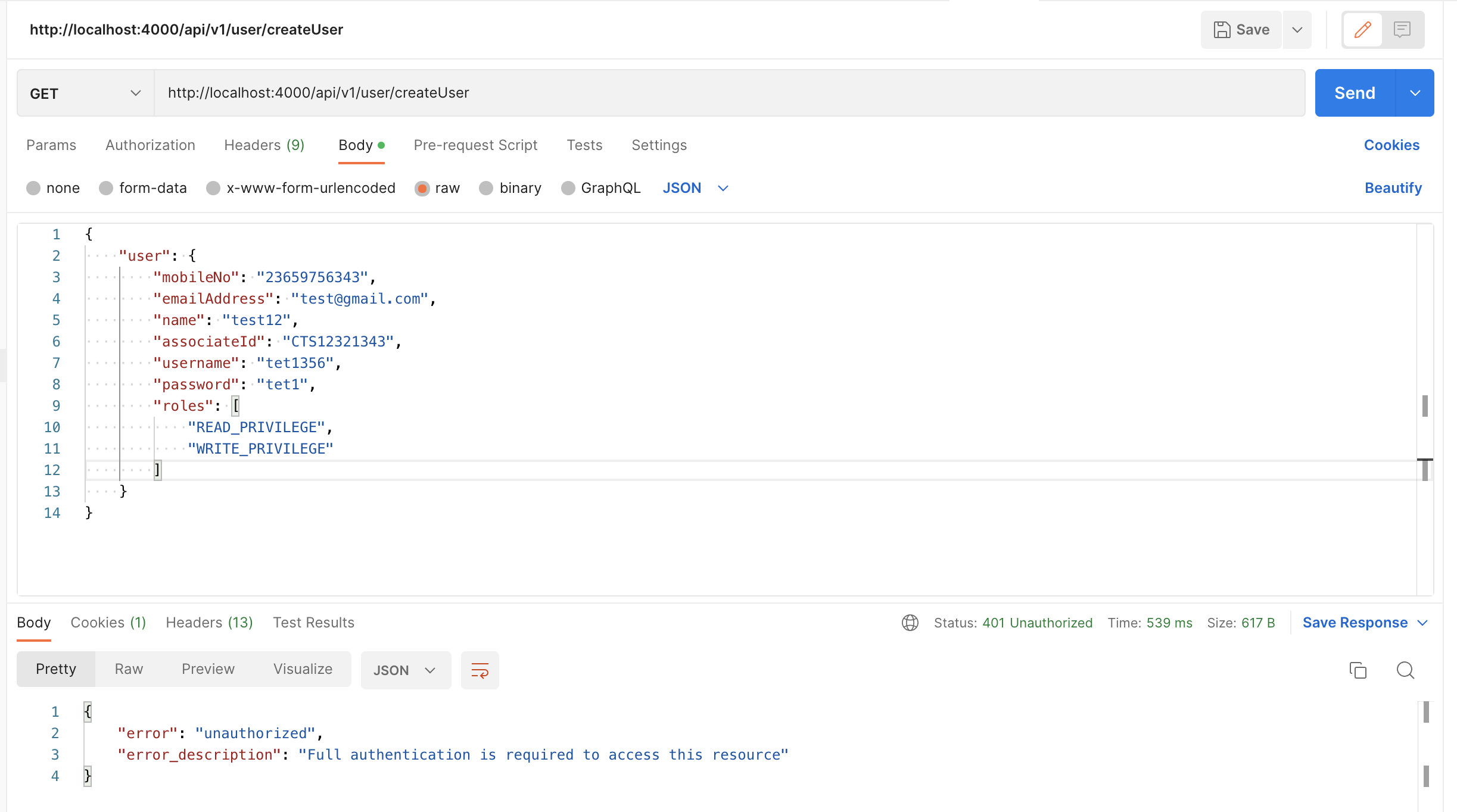
Run all the microservices. While starting the skill-tracker-query, it create skilltracker database in mysql DB and the necessary tables.

Text, letter

Description automatically generated

All the requests hit from Spring Cloud gateway which is running on the port 4000

**Trying to create user without permission:**



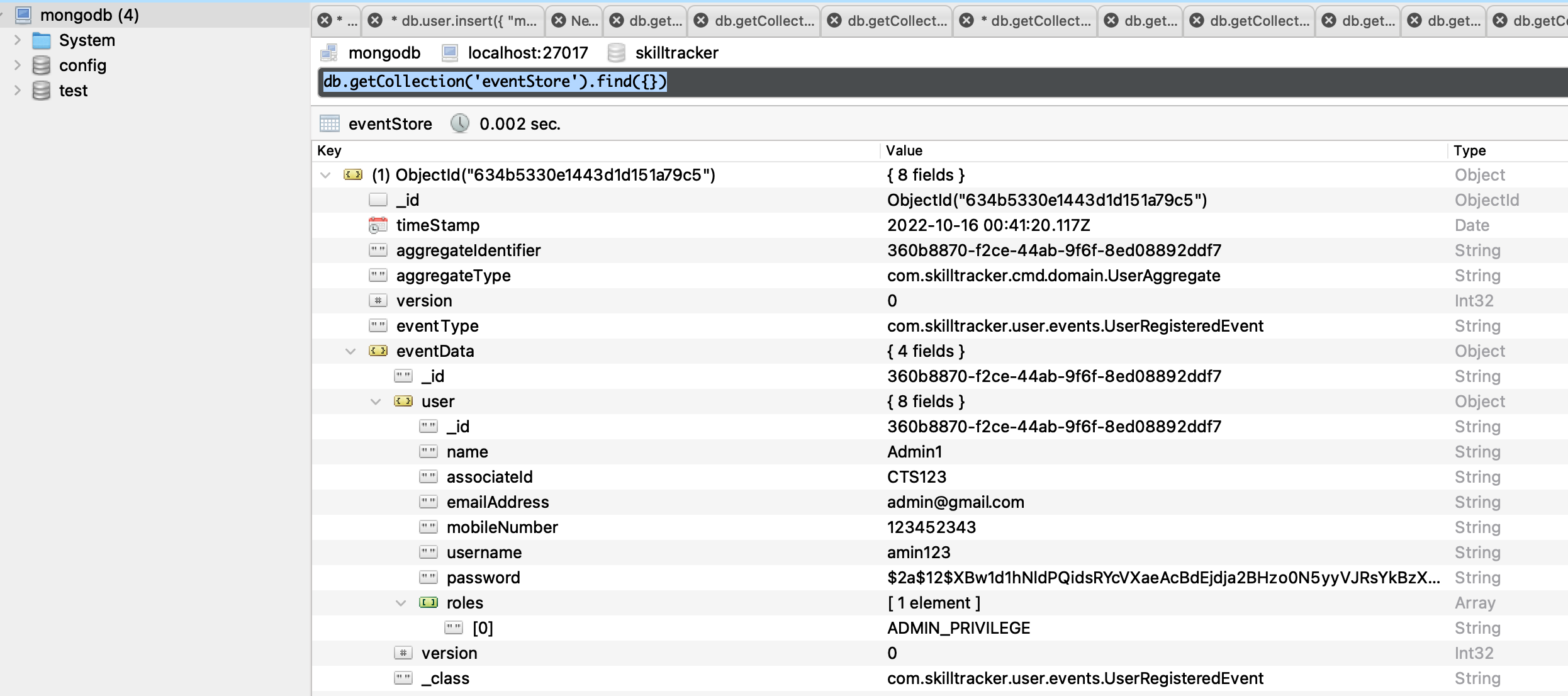
The user creation can be done only by the Admin user. Therefore, need to create Admin user first using register user API.

Graphical user interface, text, application, email

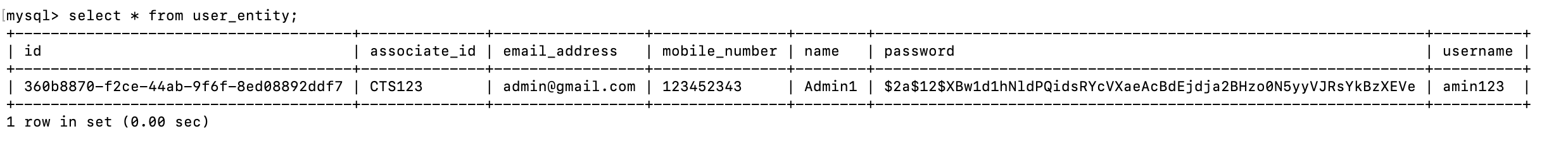
Description automatically generated

The admin user successfully created and stored in the mongoDB & mysql DB.

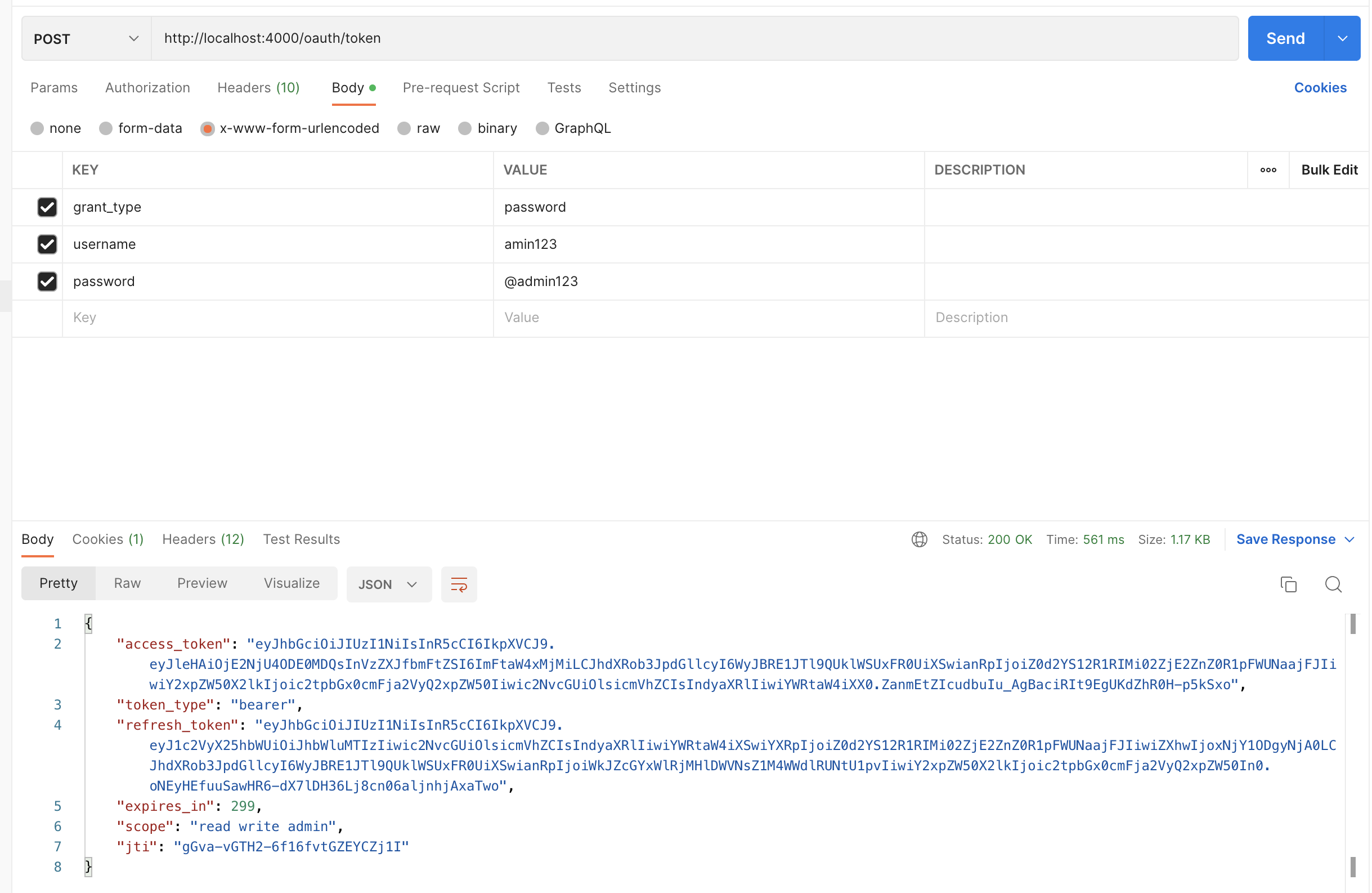
MongoDB acts as eventstore which store all events.



The same data stored in mysql:



User the admin’s username and password and hit oauth service to generate jwt to create further user.



Take the access token from the response and use it to create user with required privileges.

Update the bearer token in the authorization tab.

Graphical user interface, text, application, email

Description automatically generated

Now try to create user and it is successfully created,

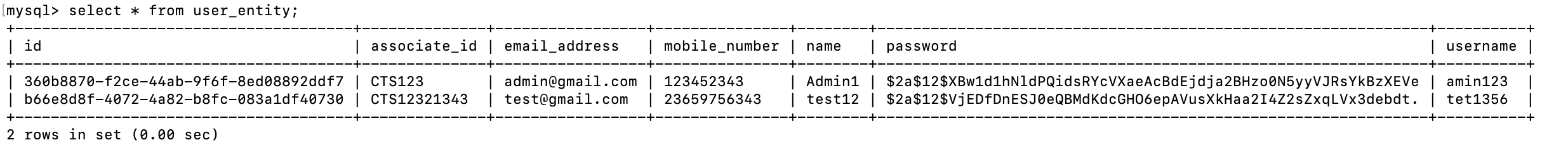
Graphical user interface, text, application, email

Description automatically generated

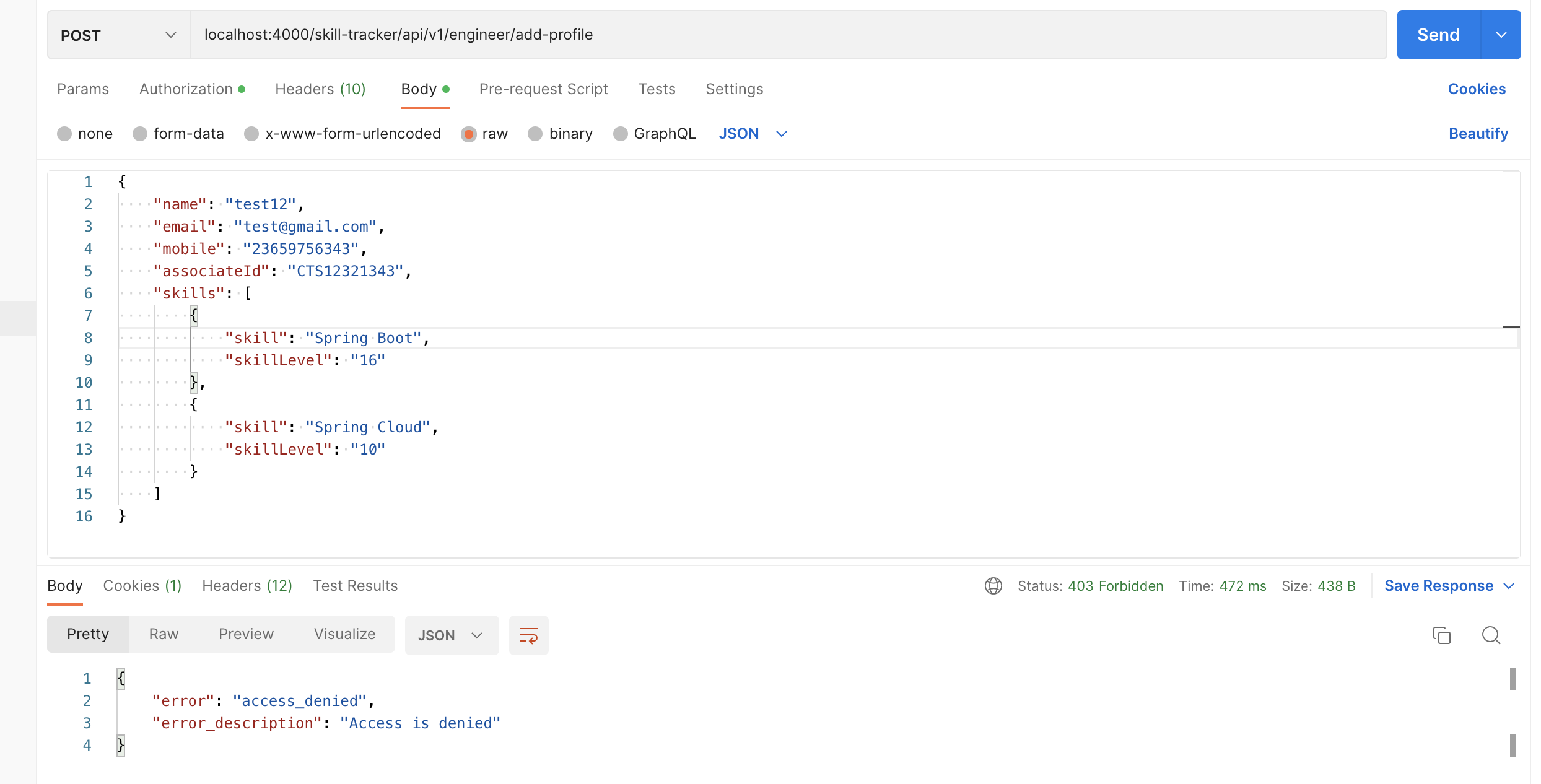
The event captured correctly in mongodb as well as mysql.

Graphical user interface, text, application, table

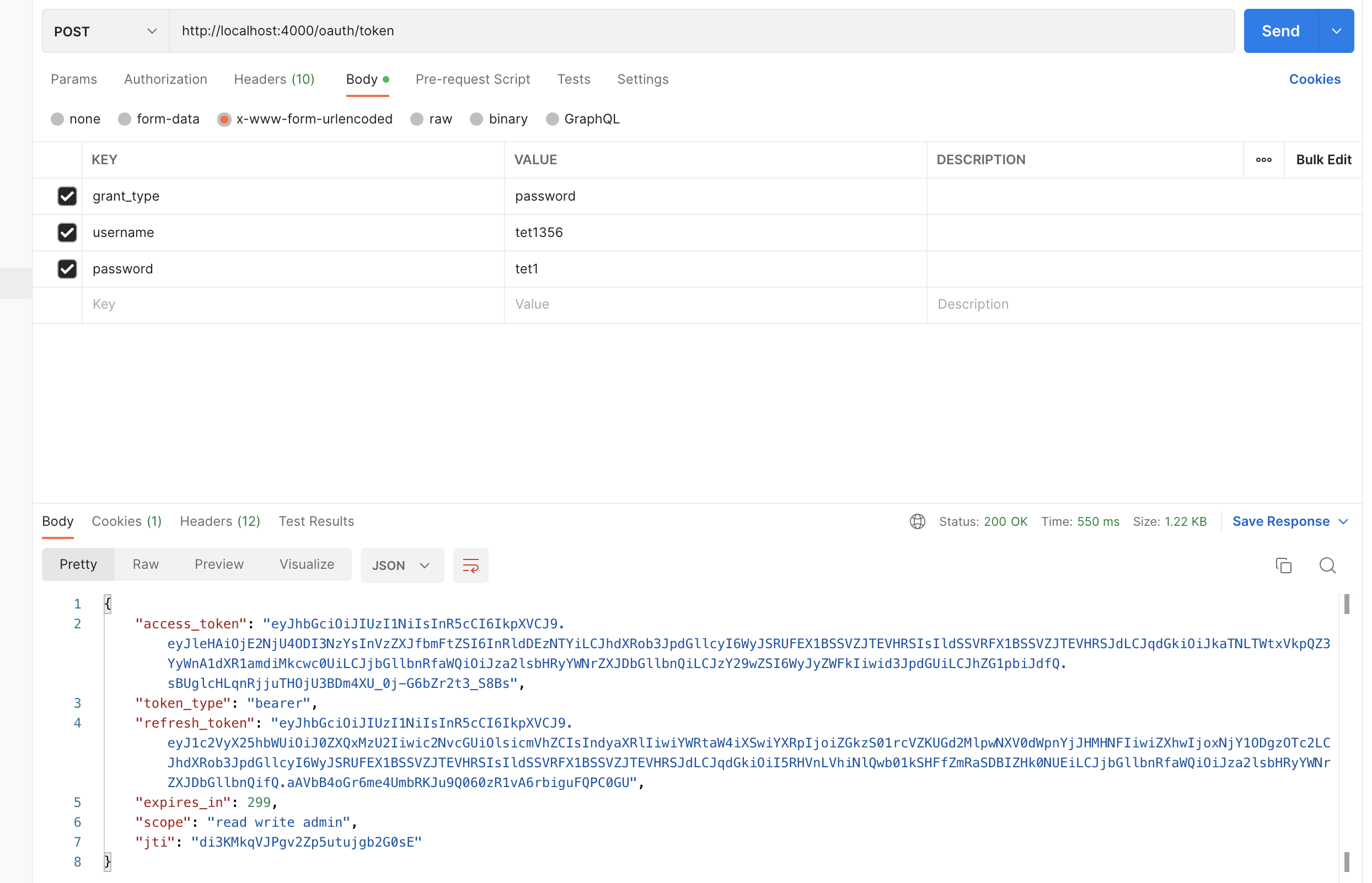
Description automatically generated



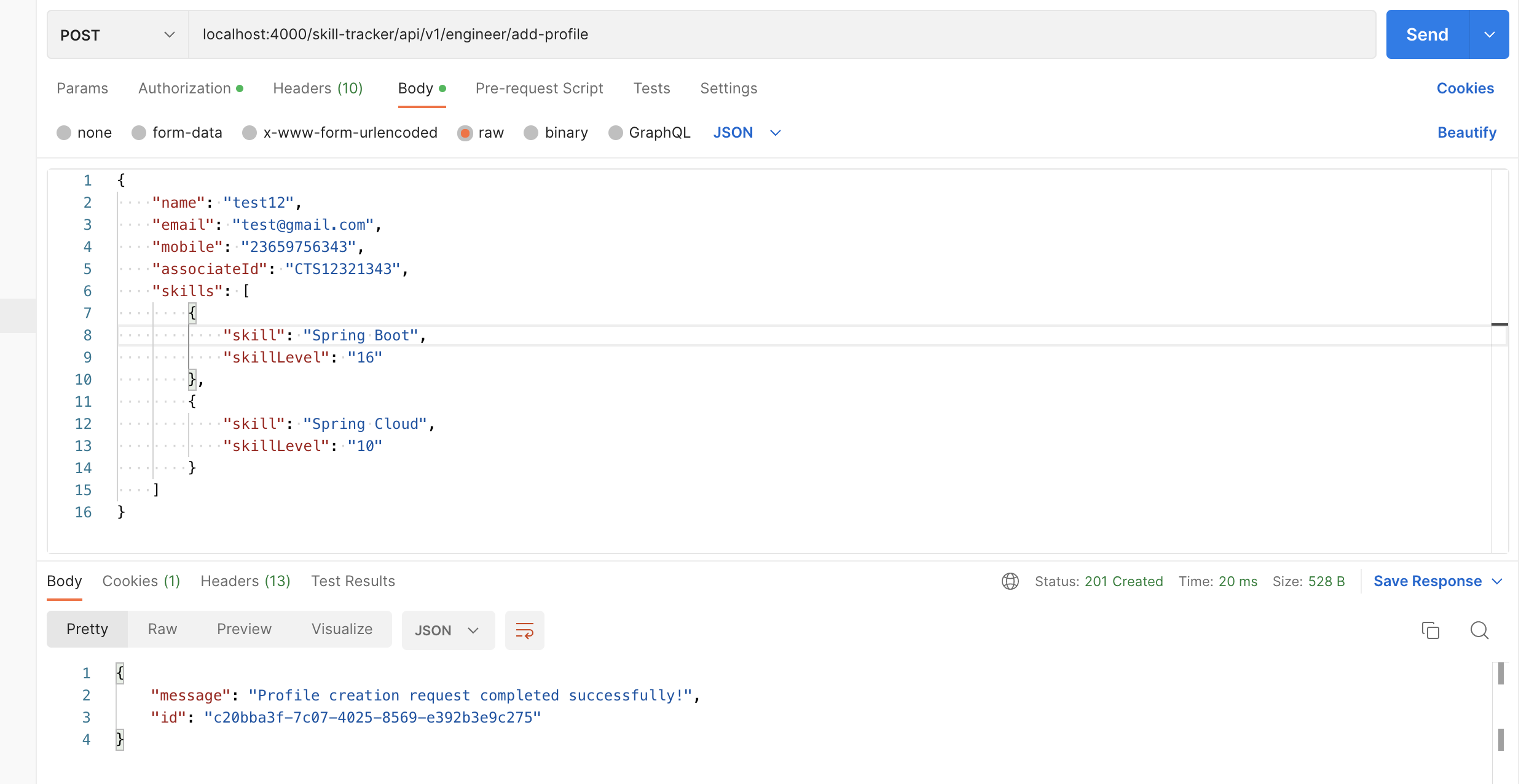
Admin user trying to create profile for the existing user which is not allowed therefore, the application throws access denied.



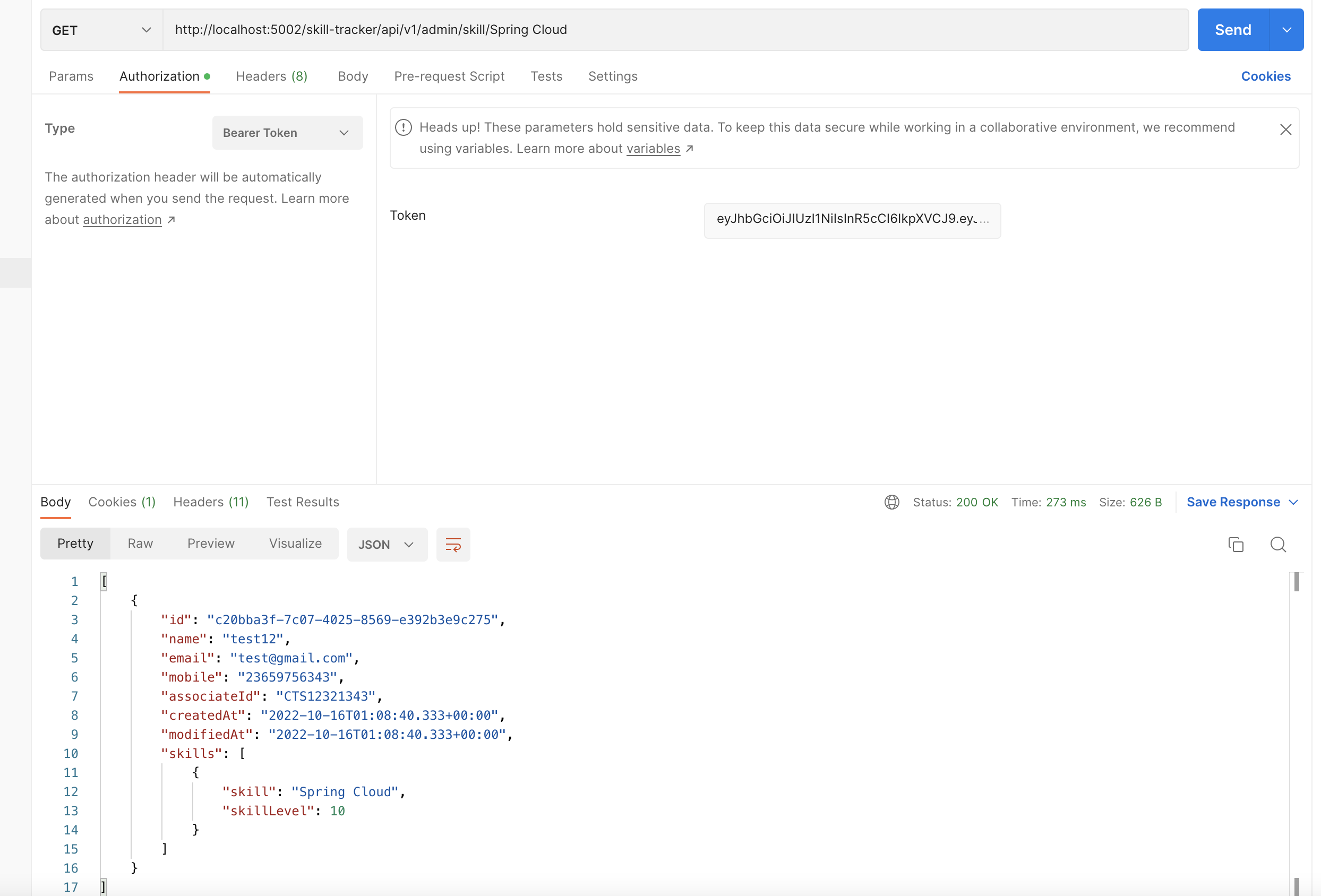
Create JWT for the user which is already created,



Use the JWT to create the profile, it is created the profile successfully.



**Query the profile based on skill name:**



**Query data based on associate id:**

Graphical user interface, text, application, email

Description automatically generated

**Query based on name:**

Graphical user interface, text, application, email

Description automatically generated