OEMS - Online Event Management System

Multi Module Repo Link: <https://github.com/jikka6303/oems-multi-module>

Submitted by:  [Jishnu Raj](mailto:2023mt93404@wilp.bits-pilani.ac.in) – 2024mt93404

# Application Description

* The application is a microservices-based system for managing users, events, and payments.
* It consists of multiple services, including an event service and a user service, which communicate with each other using REST APIs.
* The application uses Spring Boot for building the services and integrates with various repositories for data persistence and external service communication.

# Architecture

The application follows a microservices architecture with the following components:

* Event Service: Manages event-related operations such as retrieving and updating events.
* User Service: Manages user-related operations, including registering for events, adding payments, and retrieving user-specific data.
* Payment Service: Handles payment-related operations, such as processing payments and updating payment status.
* Repositories: Interfaces for communication with external services and databases.
* Controllers: REST controllers for handling HTTP requests and responses.
* Services: Business logic for handling operations related to users, events, and payments.

# Security

The application uses JWT (JSON Web Tokens) for authentication and authorization. Here's an explanation of how the security layer is implemented to authenticate the services:  
Security Configuration

1. JWT Token Generation - [Source](https://github.com/jikka6303/oems-user-service/blob/33c54681f24829bb26bbe16c4b6c23dbbc00c5f2/src/main/java/com/bits/oems/user/utility/AppJwtUtility.java#L38):

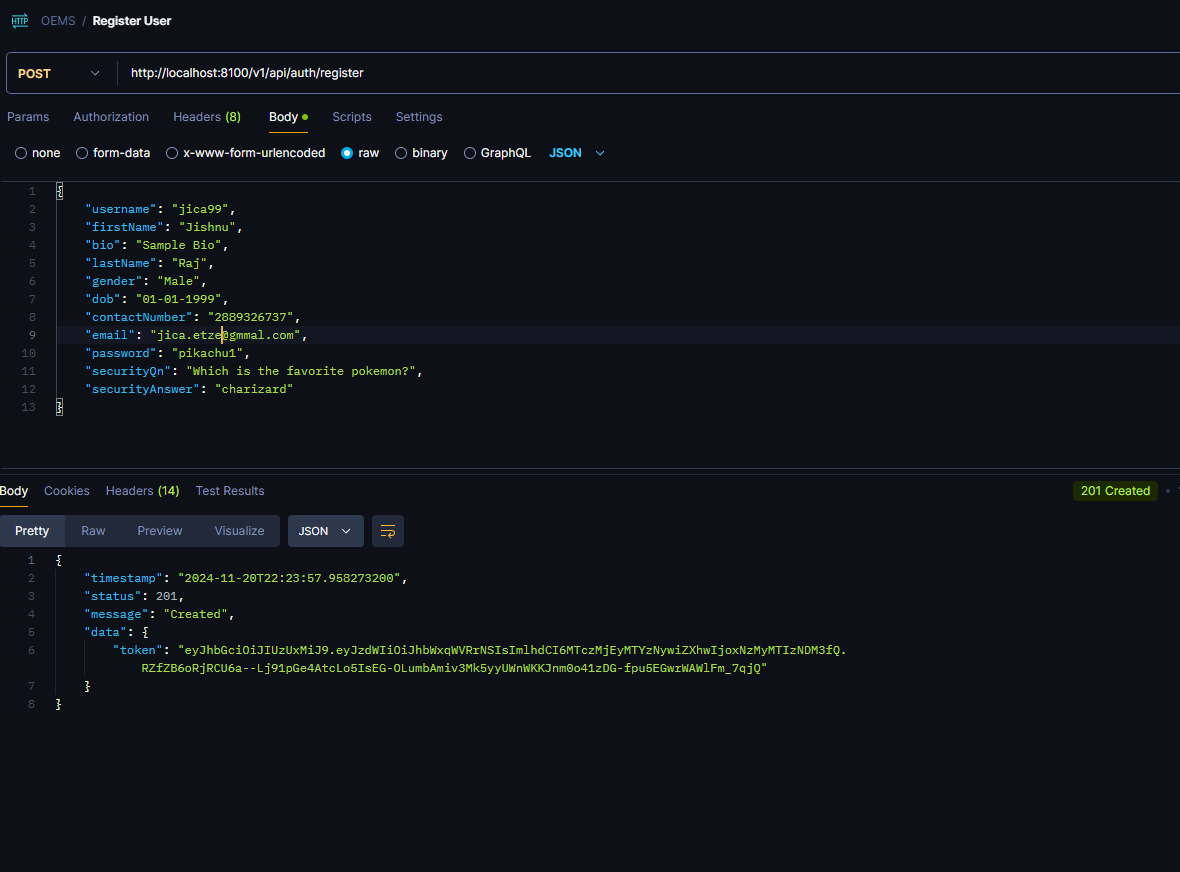
When a user logs in, the application generates a JWT token that contains the user's information and expiration time. This token is signed using a secret key to ensure its integrity.

1. JWT Token Validation - [Source](https://github.com/jikka6303/oems-user-service/blob/33c54681f24829bb26bbe16c4b6c23dbbc00c5f2/src/main/java/com/bits/oems/user/service/impl/JWTValidationService.java#L29):

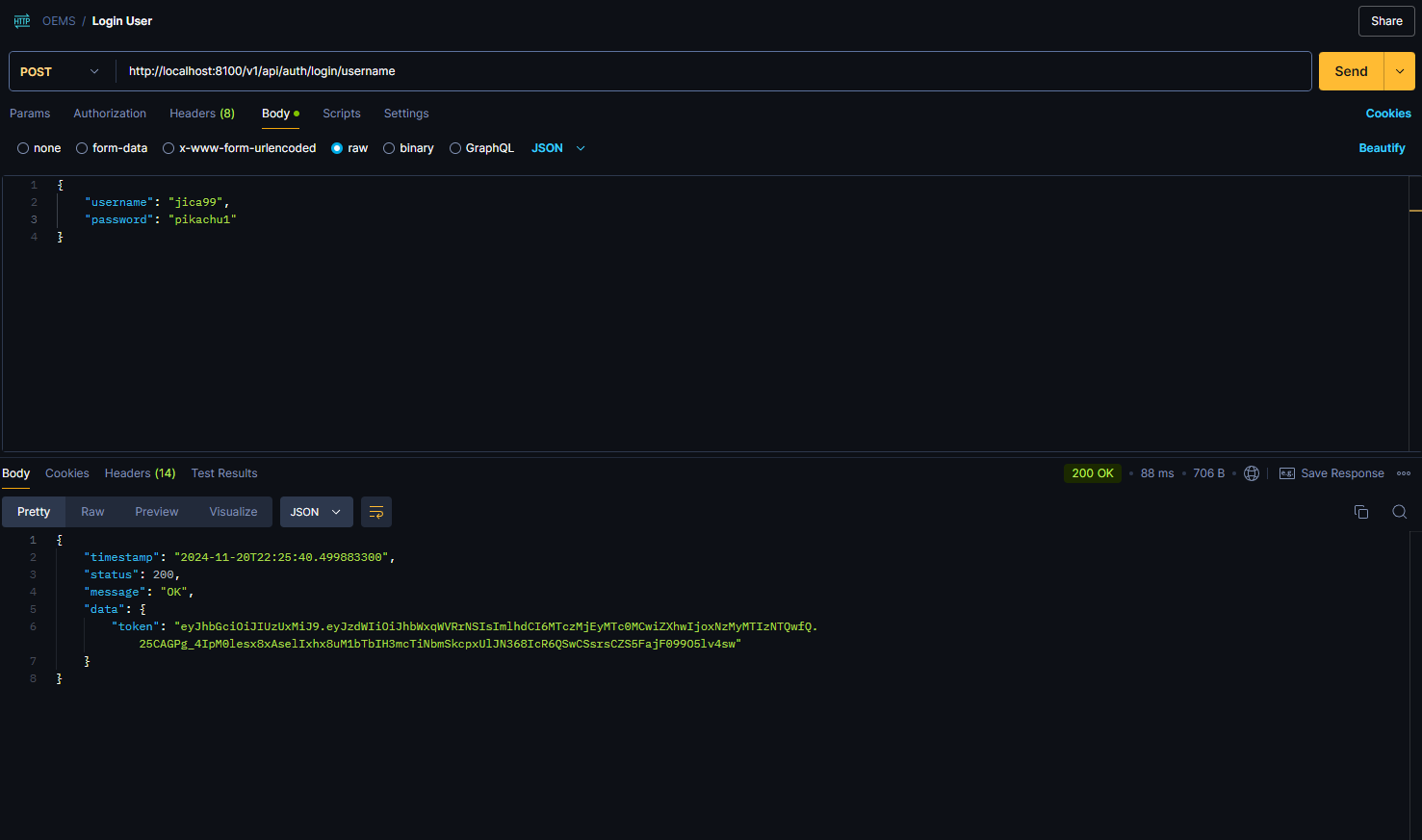
For each incoming request, the application extracts the token from the Authorization header. The token is then validated to check its signature, expiration, and claims. If the token is valid, the user is authenticated and granted access to the requested resource.

# Screenshots

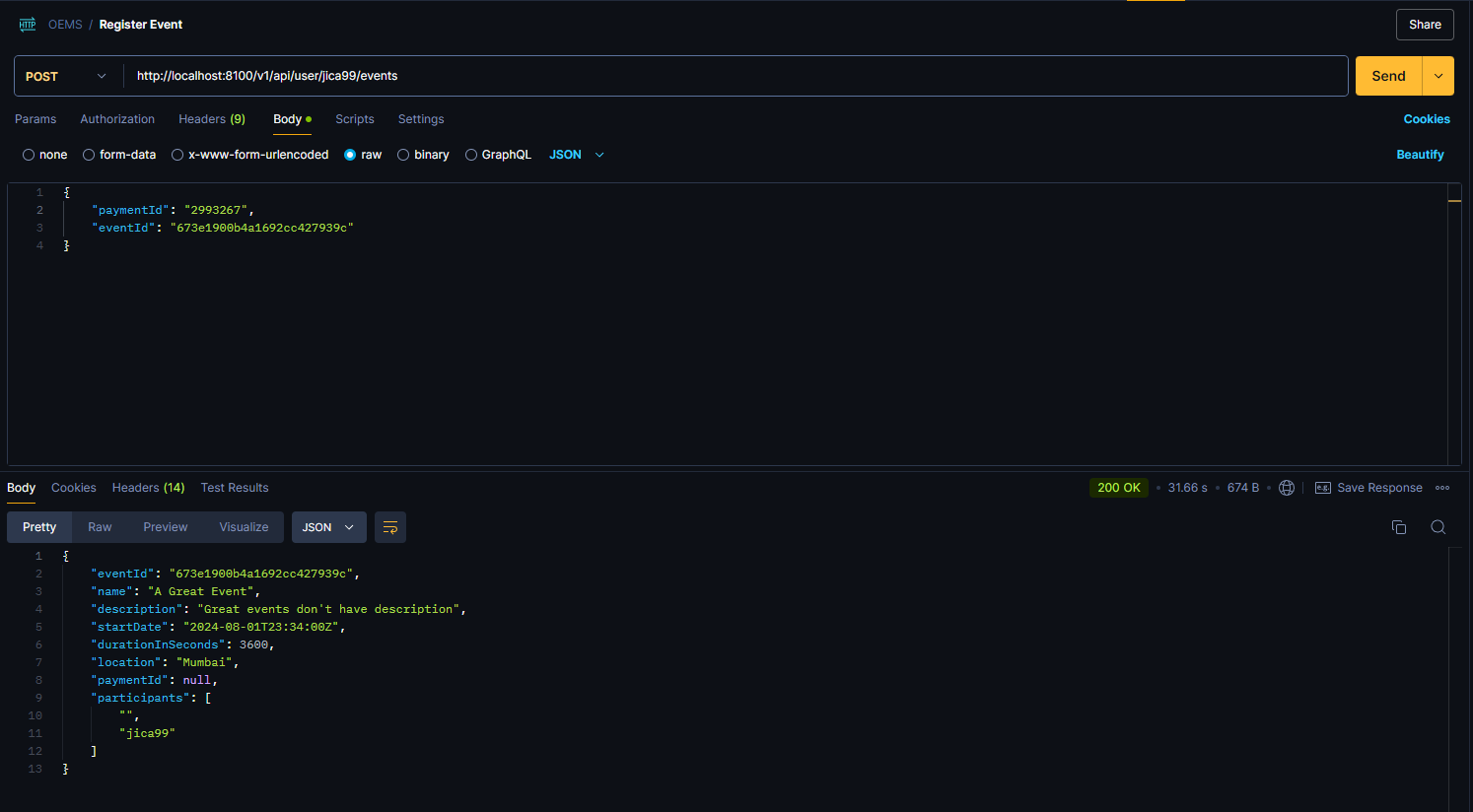
* Register User



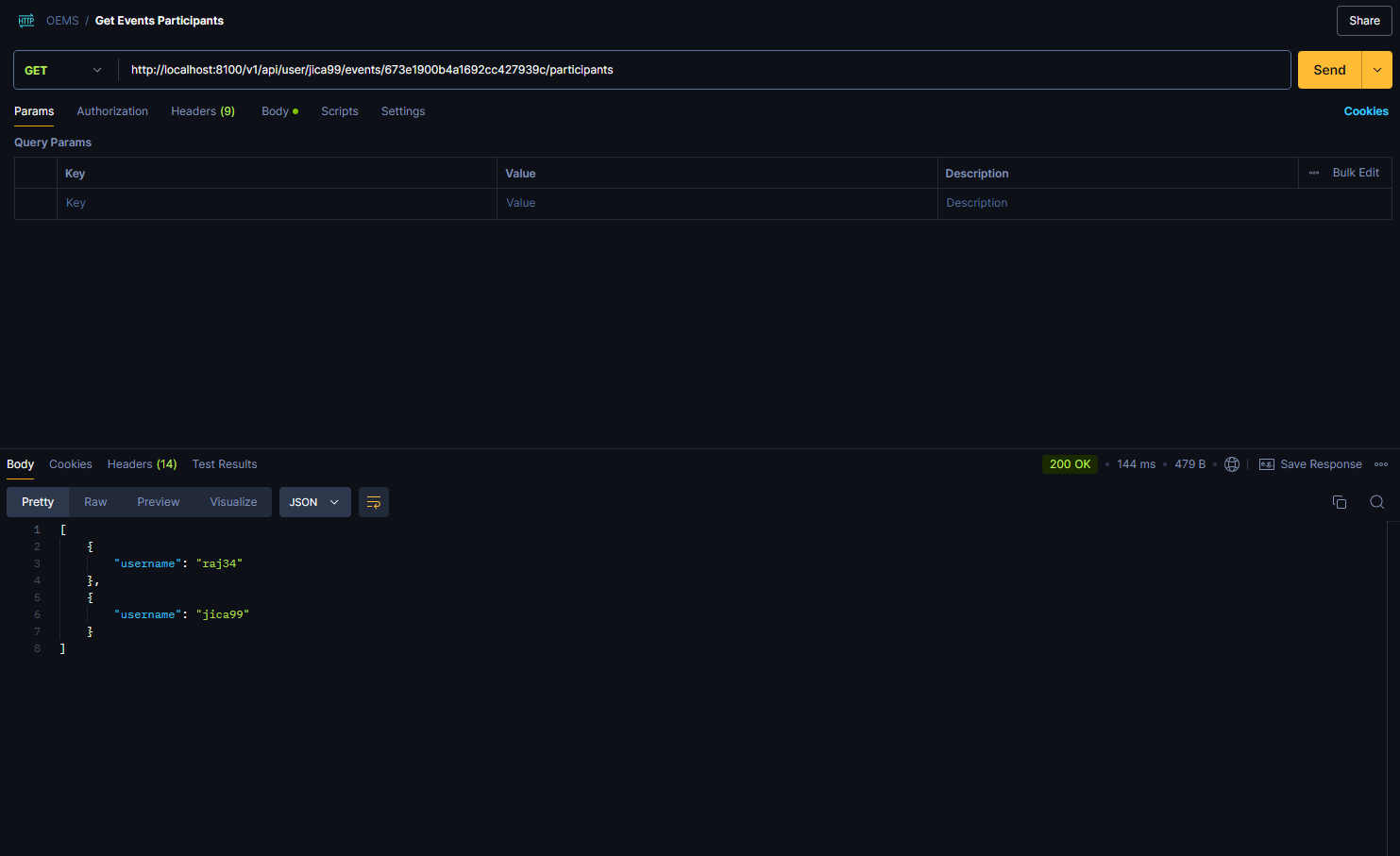
* Login User



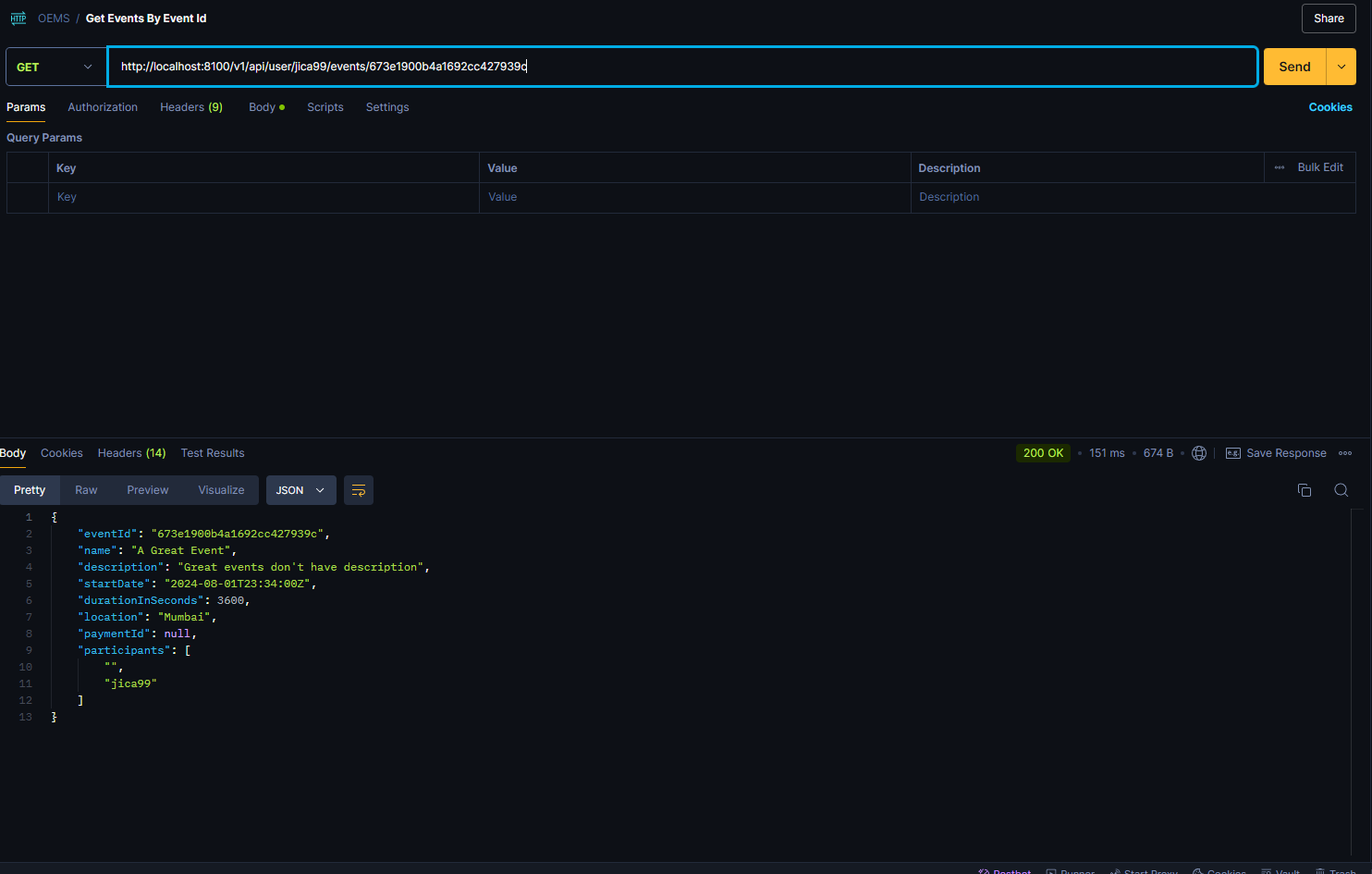
* Register Event



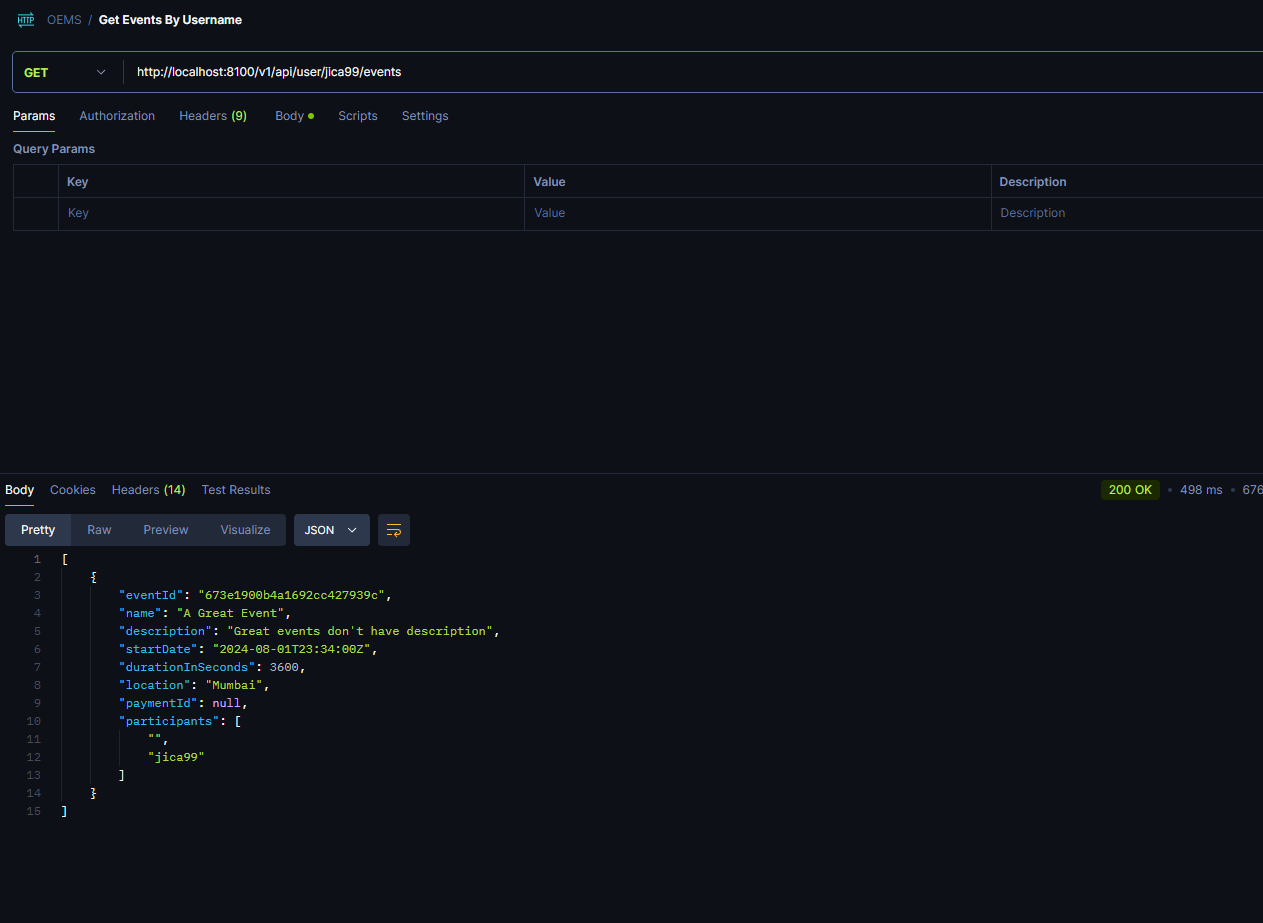
* Get Event Participants



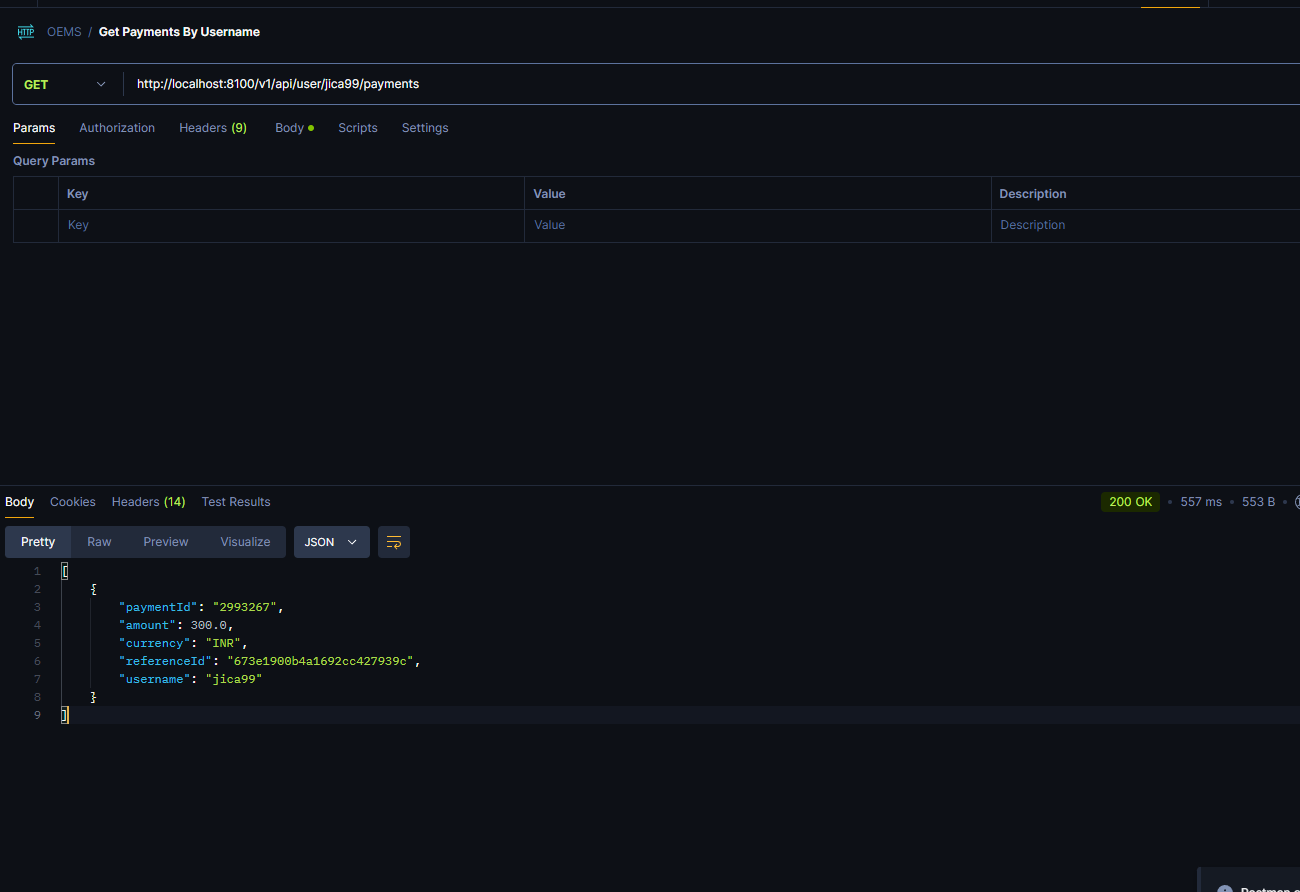
* Get Events By Event ID



* Get Events By Username



* Get Payments By Username



* Make Payment

