

# Backend Intern Assignment

**GrowthX**

**By Gaurav Raturi**

## Table of Contents

1. [Introduction](#)
2. [Getting Started](#)
3. [Authentication](#)
4. [Endpoints](#)
  - [User Endpoints](#)
  - [Admin Endpoints](#)
5. [Error Handling](#)

## Introduction

This documentation provides detailed information about the Assignment Management API, designed to manage users, assignments, and administrative actions. The system supports user registration, login, assignment uploads, and status updates for assignments.

The API is built using Spring Boot and MongoDB and follows RESTful principles for ease of integration.

---

## Getting Started

### Base URL

- <http://localhost:8080> (Default for local development)

### Features

- **User Registration and Login:** Allows users and admins to register and log in securely.
  - **Assignment Management:** Enables users to upload assignments and view their status.
  - **Admin Features:** Allows admins to fetch and update assignment statuses.
  - **Role-based Access:** Differentiates between USER and ADMIN roles.
- 

## Technologies Used

- **Programming Language:** Java
- **Backend Framework:** Spring Boot
- **Database:** MongoDB
- **Security:** Spring Security (CSRF disabled for development)

### API Standards: RESTful APIs

---

## Authentication

This API uses role-based access. Admin actions require the ADMIN role, while other actions can be performed by a USER.

---

## Endpoints

### 1. User Endpoints

#### 1.1. Register User

**Endpoint:** POST /user/register

**Postman Endpoint:** http://localhost:8080/user/register

**Description:** Registers a new user or admin.

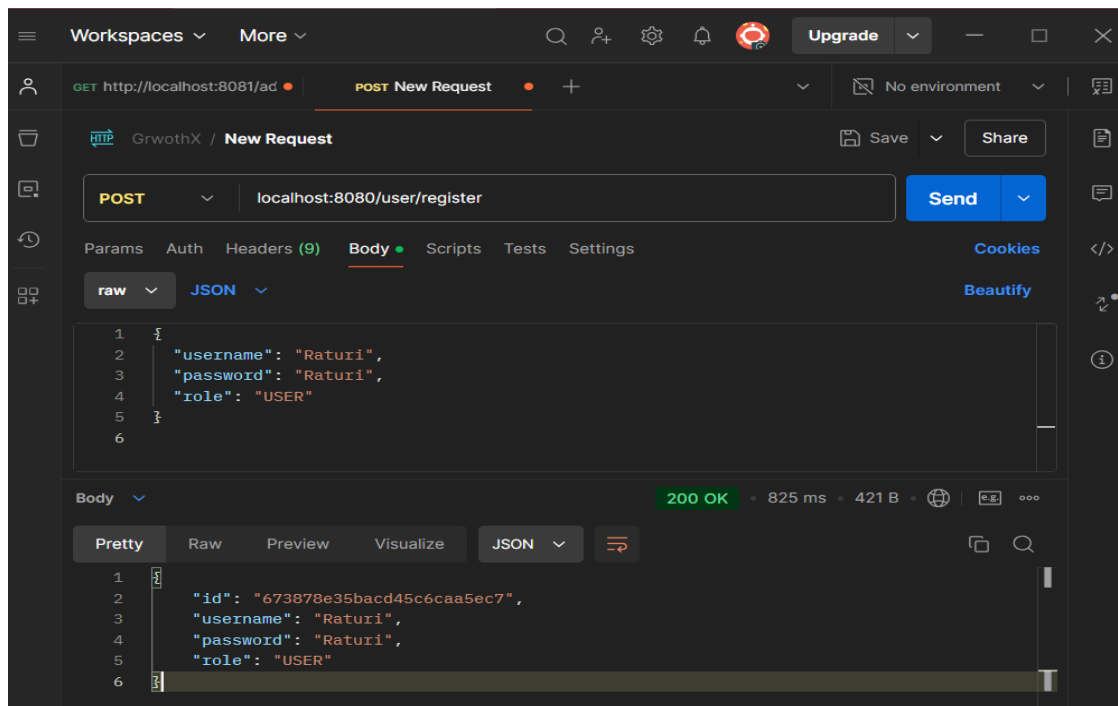
**Request Body:**

```
{  
  "username": "Raturi",  
  "password": "Raturi",  
  "role": "USER"  
}
```

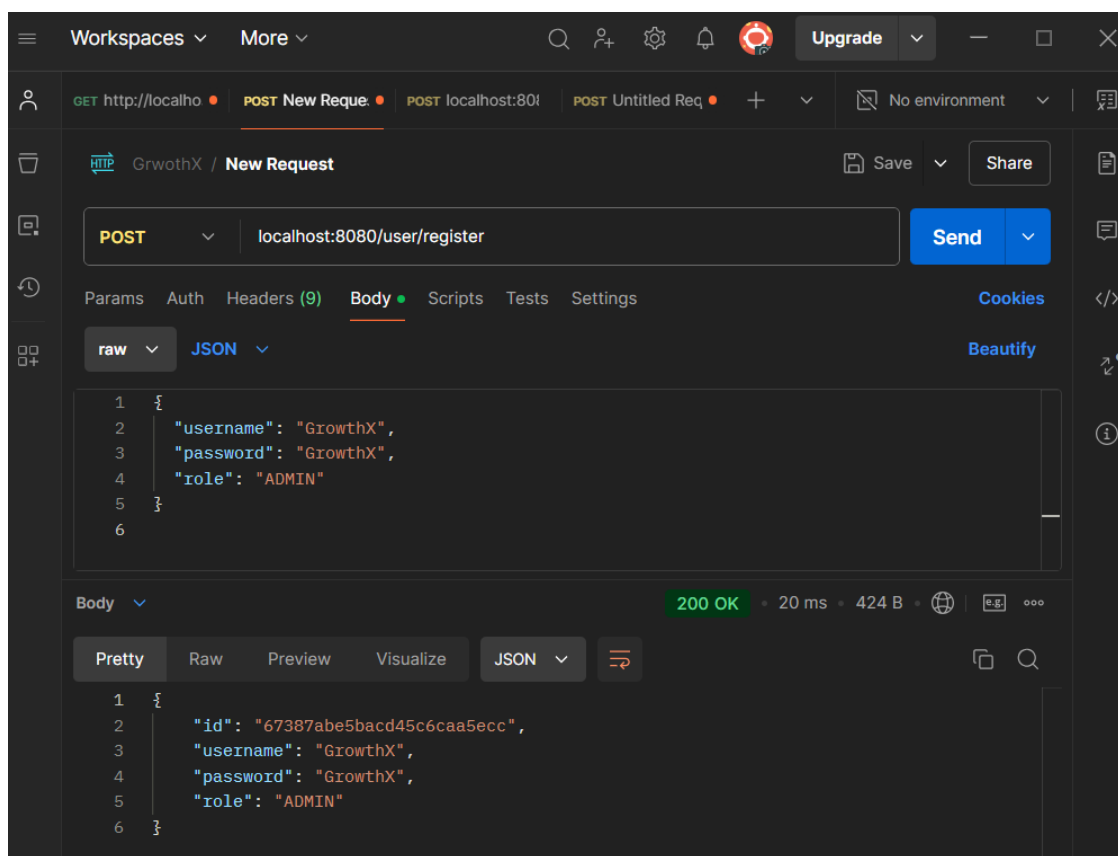
**Response:**

```
{  
  "id": "67387a375bacd45c6caa5ecb",  
  "username": "Gaurav ",  
  "role": "USER"  
}
```

### Registering Raturi as a user:



### Registering GrowthX as an admin:



## 1.2. Login

**Endpoint:** POST /user/login

**Postman Endpoint:** POST http://localhost:8080/user/login

**Description:** Authenticates a user or admin.

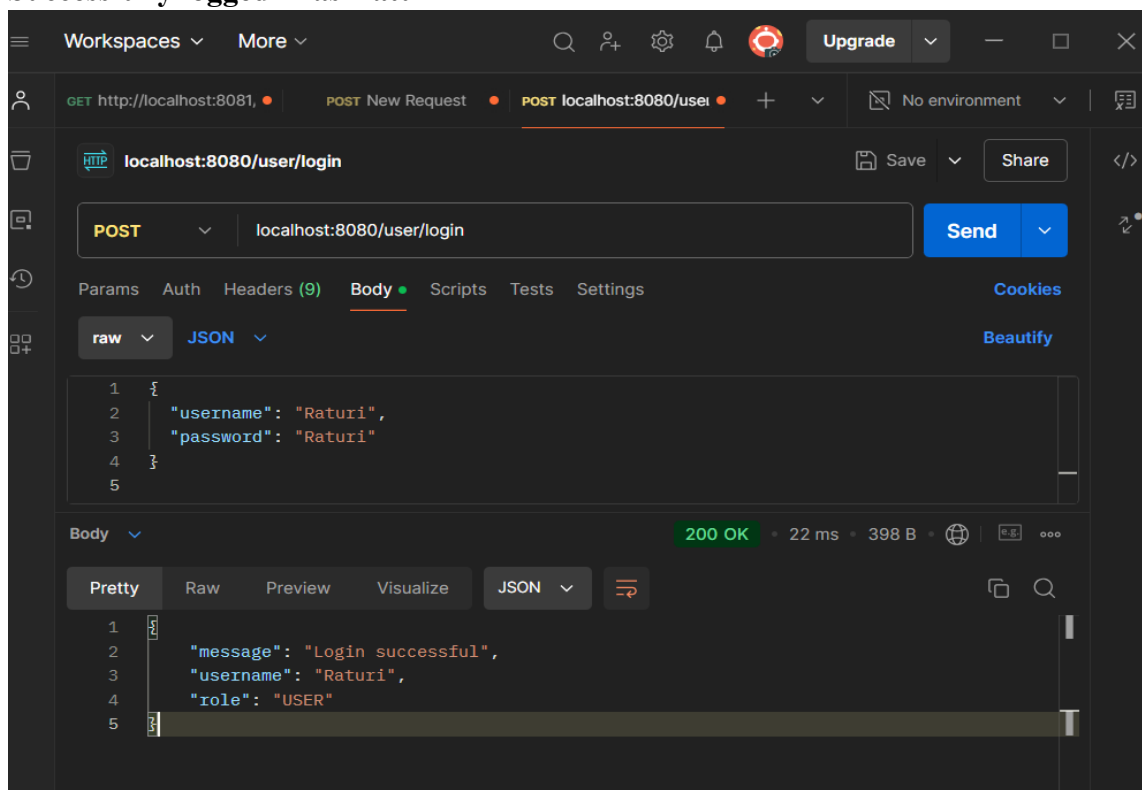
**Request Body:**

```
{
  "username": "Raturi",
  "password": "Raturi"
}
```

**Response:**

```
{
  "message": "Login successful",
  "username": "Raturi",
  "role": "USER"
}
```

**Successfully logged in as Raturi**



### 1.3. Upload Assignment

**Endpoint:** POST /user/upload

**Description:** Allows a user to upload an assignment.

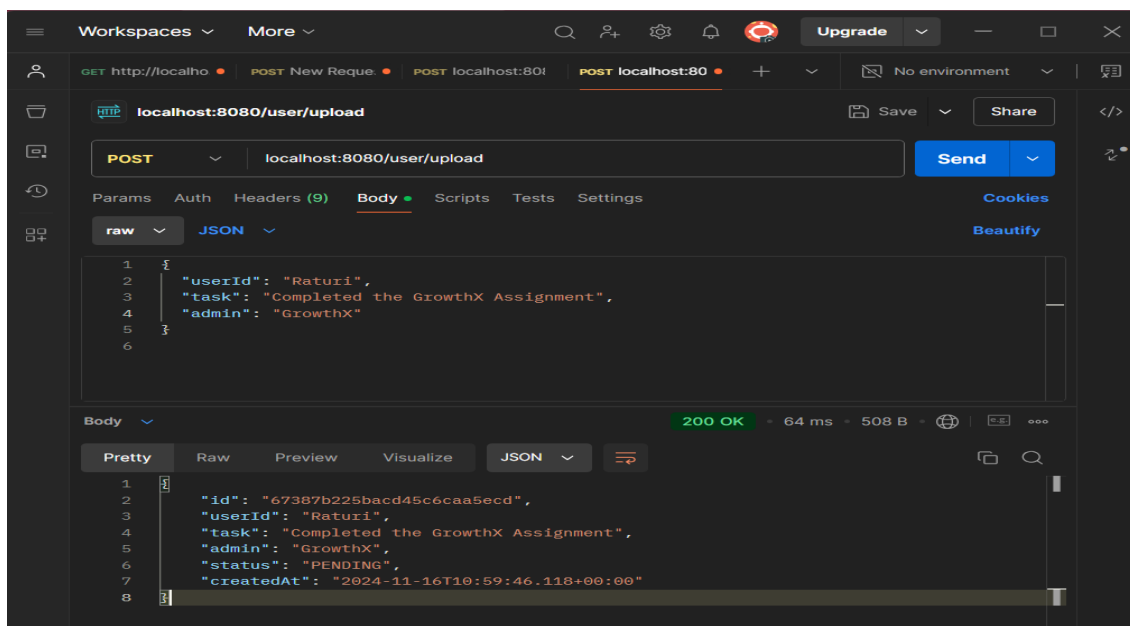
**Request Body:**

```
{
  "userId": "Raturi",
  "task": "Completed the GrowthX Assignment",
  "admin": "GrowthX"
}
```

**Response:**

```
{
  "id": "67387b225bacd45c6caa5ecd",
  "userId": "Raturi",
  "task": "Completed the GrowthX Assignment",
  "admin": "GrowthX",
  "status": "PENDING",
  "createdAt": "2024-11-16T10:59:46.118+00:00"
}
```

**Uploading the assignment for user Raturi and assigning it to the admin GrowthX.**



## 1.4. Get User Assignments

**Endpoint:** GET /user/assignments?userId=user\_id

**Postman Endpoint:** http://localhost:8080/user/assignments?userId=Raturi

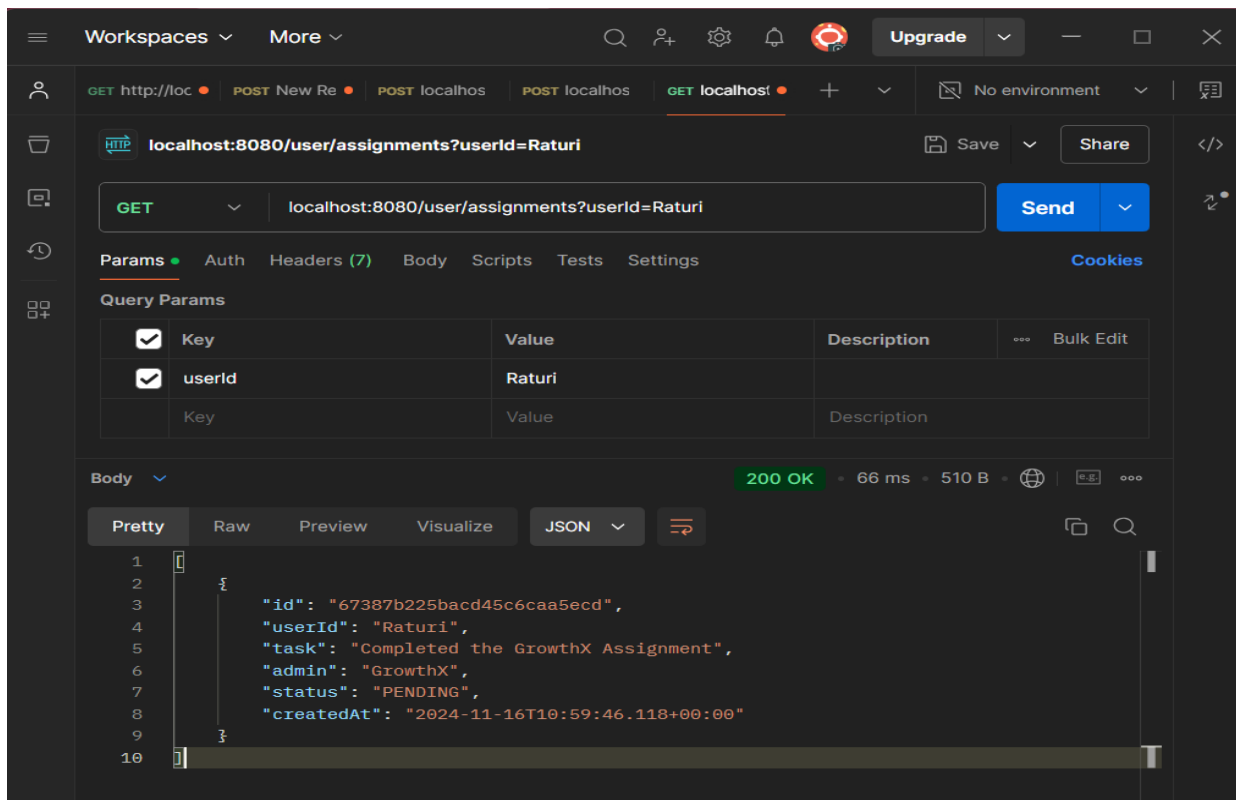
**Description:** Fetches all assignments for a user.

**Query Parameter:** userId (String): ID of the user.

**Response:**

```
[
  {
    "id": "67387b225bacd45c6caa5ecd",
    "userId": "Raturi",
    "task": "Completed the GrowthX Assignment",
    "admin": "GrowthX",
    "status": "PENDING",
    "createdAt": "2024-11-16T10:59:46.118+00:00"
  }
]
```

Fetching all assignments submitted by user Raturi.





## 1.5. Get All Admins

**Endpoint:** GET /user/admins

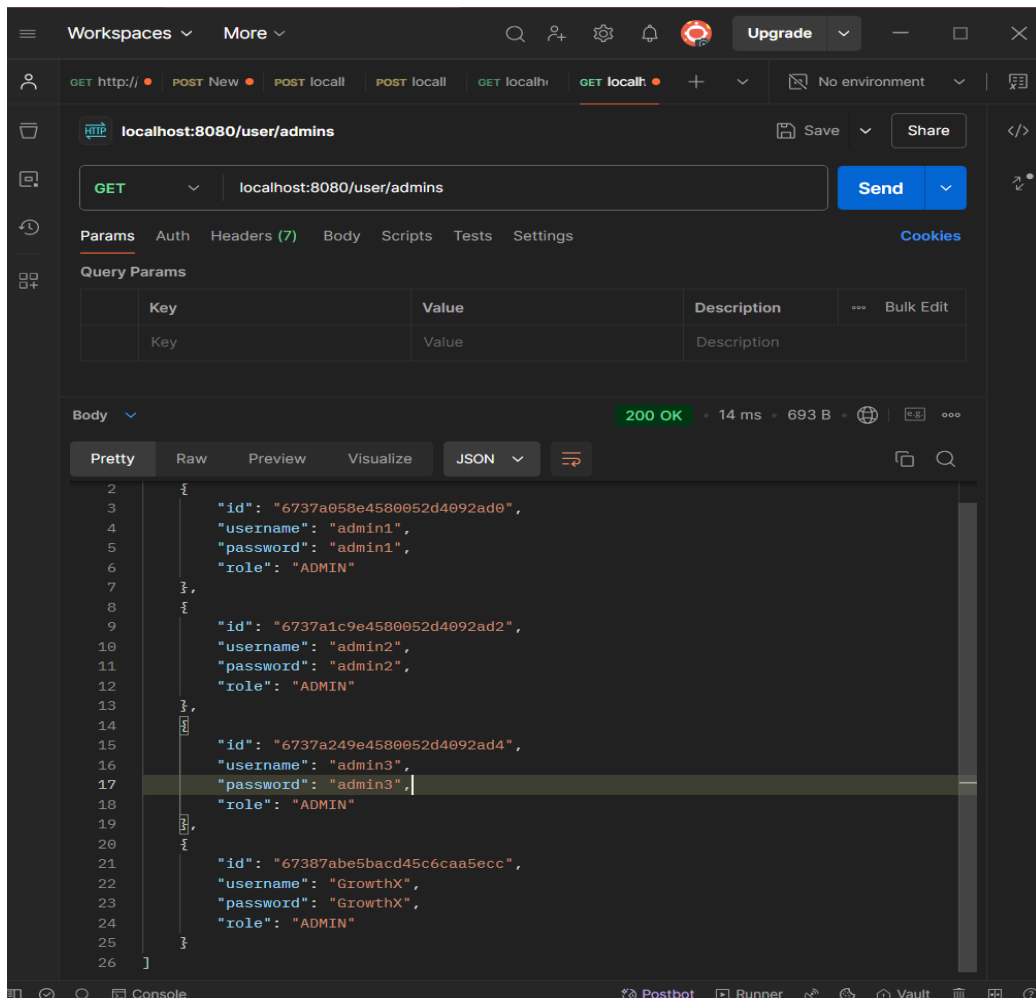
**Postman Endpoint:** localhost:8080/user/admins

**Description:** Fetches all users with the ADMIN role.

**Response:**

```
{
  {
    "id": "67387abe5bacd45c6caa5ecc",
    "username": "GrowthX",
    "password": "GrowthX",
    "role": "ADMIN"
  }
}
```

Fetching all users with the ADMIN role.



## 2. Admin Endpoints

### 2.1. Get Assignments for Admin

**Endpoint:** GET /admin/assignments

**Postman Endpoints:**

http://localhost:8080/admin/assignments?adminUsername=GrowthX

**Description:** Fetches all assignments assigned to a specific admin.

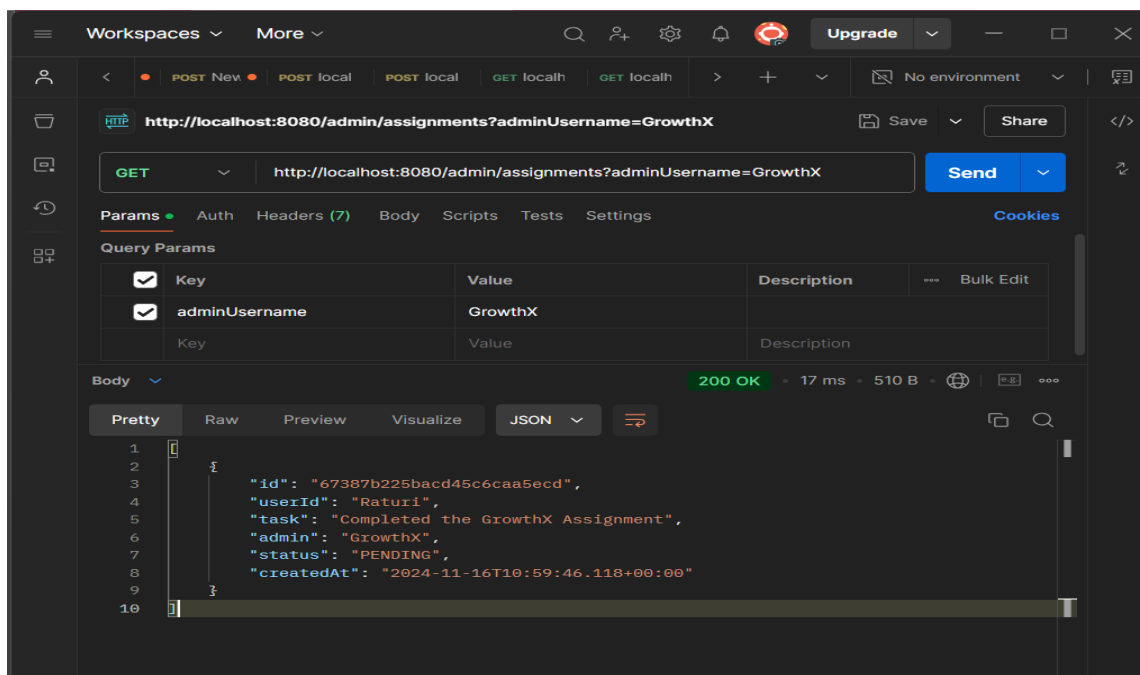
**Query Parameter:**

- adminUsername (String): Username of the admin.

**Response:** [

```
{
  "id": "67387b225bacd45c6caa5ecd",
  "userId": "Raturi",
  "task": "Completed the GrowthX Assignment",
  "admin": "GrowthX",
  "status": "PENDING",
  "createdAt": "2024-11-16T10:59:46.118+00:00"
}]
```

Fetching all assignments assigned to GrowthX.



## 2.2. Update Assignment Status

**Endpoint:** POST /admin/assignments/{id}/{action}

**Postman Endpoint:**

http://localhost:8080/admin/assignments/67387b225bacd45c6caa5ecd/accept

**Description:** Updates the status of an assignment.

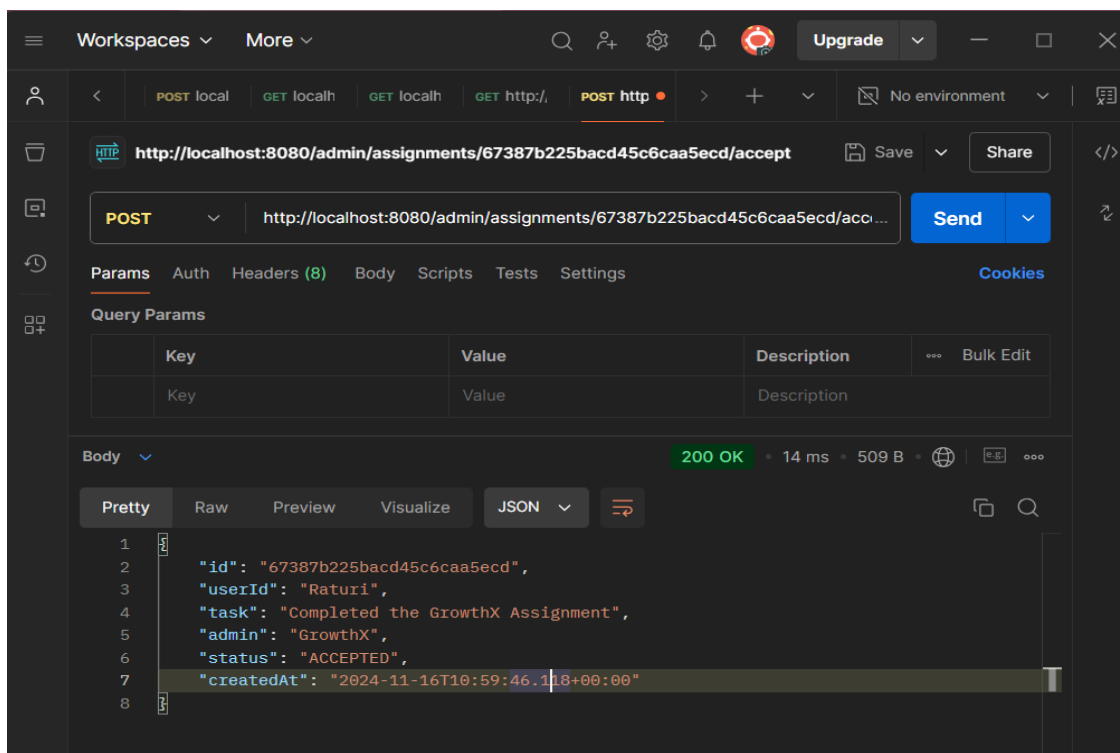
**Path Parameters:**

- id (String): ID of the assignment.
- action (String): Action to perform (accept or reject).

**Response:**

```
{
  "id": "67387b225bacd45c6caa5ecd",
  "userId": "Raturi",
  "task": "Completed the GrowthX Assignment",
  "admin": "GrowthX",
  "status": "ACCEPTED",
  "createdAt": "2024-11-16T10:59:46.118+00:00"
}
```

**Updating the status of the assignment for user Raturi to "Accepted."**



## Passing the wrong Assignment Id:

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/admin/assignments/123456/accept`. The response is a **404 Not Found** status, indicating that the assignment with ID 123456 does not exist.

**Query Params**

Key	Value	Description
Key	Value	Description

**Body**

1 Assignment not found with ID: 123456

## Error Handling

### HTTP Status Code Description

400	Bad Request. Invalid input or format.
401	Unauthorized. Invalid credentials.
404	Not Found. Resource not available.
500	Internal Server Error.