### **1. Overview**

**Project Name**: CRUD API with Express and PostgreSQL

**Description**: This API allows users to insert and read student’s information in a PostgreSQL database. It’s built using Express.js and PostgreSQL.

**Base URL**: <http://localhost:3000>

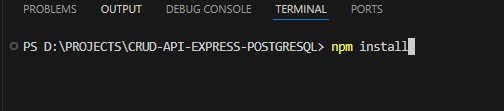
### **2. Setup Instructions**

**Prerequisites**:

* Node.js
* PostgreSQL
* npm (Node Package Manager)

**Installation**:

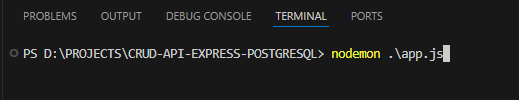
1-Install dependencies:



2- Configure PostgreSQL database:

* Create a database and a user with the necessary permissions.
* Update the database configuration in the project (usually in a .env file)

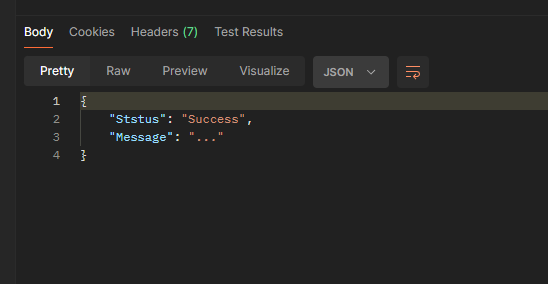
3- Start the server:



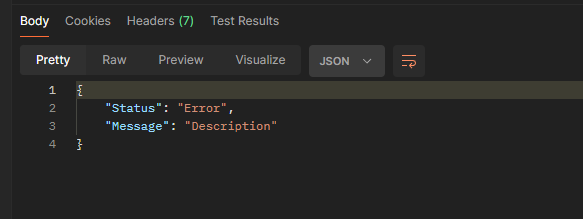
**3. API Endpoints**

#### **Common Response Format**

**Success**:

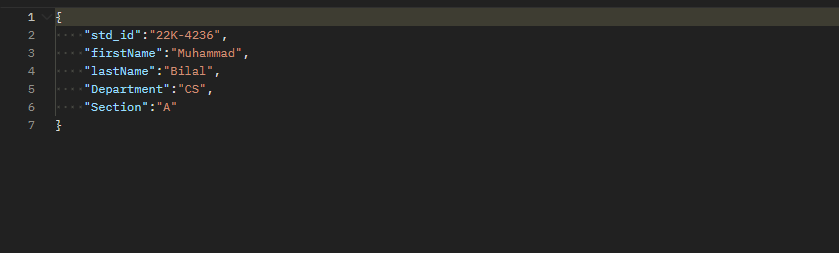


**Error:**

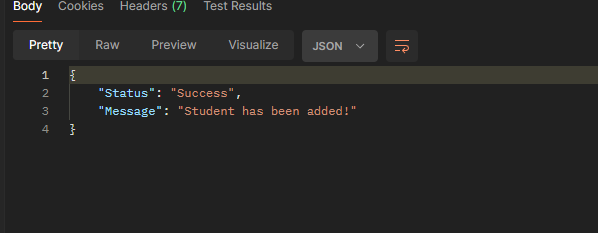


#### **1. Insert a Student**

* **Endpoint**: POST /user/add-student
* **Description**: Insert a new student.
* **Request Body:**



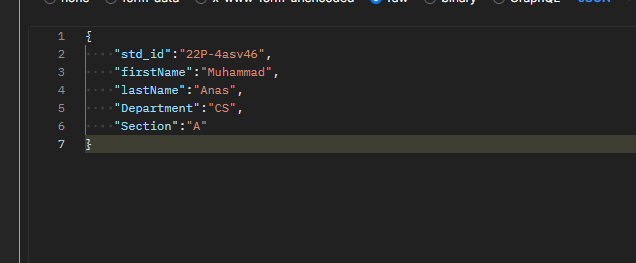
* **Response:**



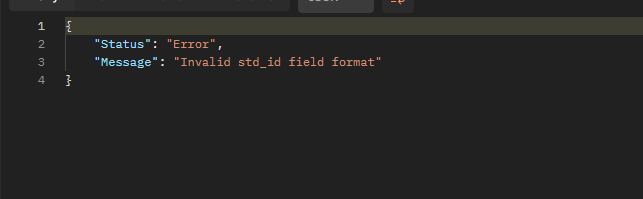
**Error Handling for Insert a student:**

* **Invalid std\_id Format**:

**Request Body**:

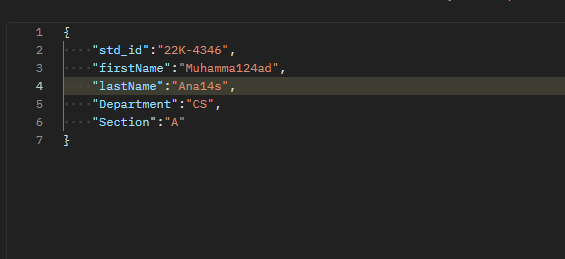


**Response:**

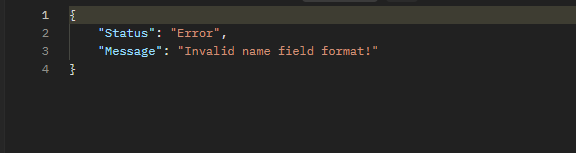


* **Invalid firstname or lastname Format:**

**Request Body:**

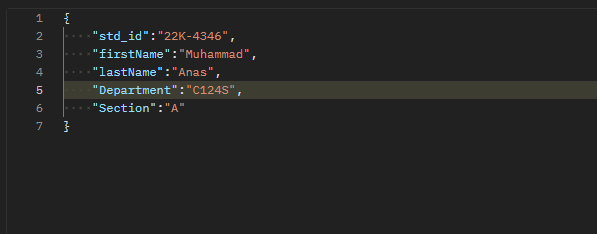


**Response:**

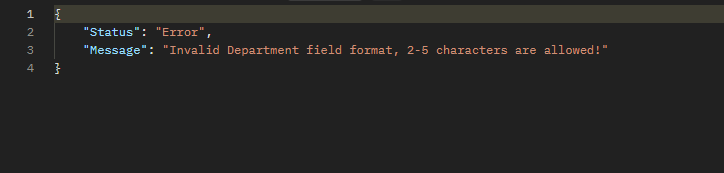


* **Invalid Department Format:**

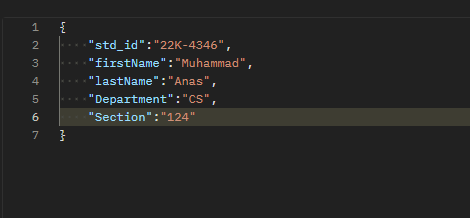
**Request Body:**



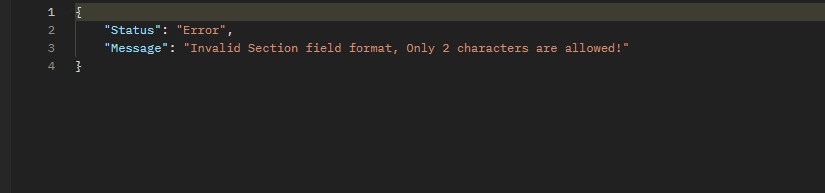
**Response:**



* **Invalid Section Format:**

**Request Body:**  


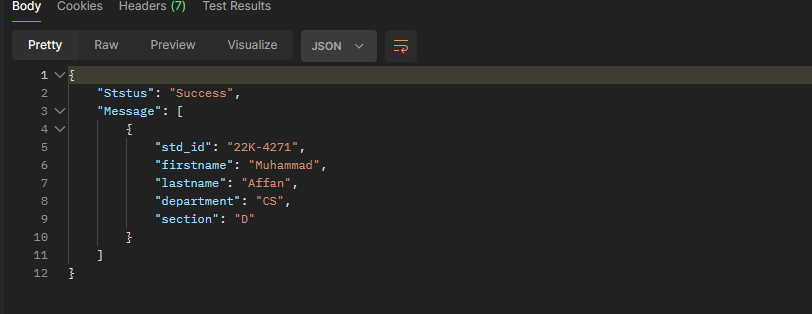
**Response:**



**2. Retrieve a Single Student’s data**

* **Endpoint**: GET /user/get-student-by-id/:id
* **Description**: Retrieve student’s information by ID.
* **Request:**

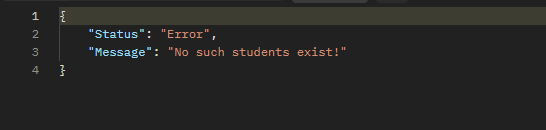


* **Response:** 
* **Invalid input std\_id Format:**

**Request Params:**



**Response:**



**4. Error Handling**

List of common error responses for each endpoint, such as:

**400 Bad Request**:

* Invalid input data.
* Missing required fields.

**404 Not Found**:

* Student not found for the given ID.

**500 Internal Server Error**:

* Unexpected server error.