API TESTING WITH POSTMAN

INTRODUCTION

The Blood Donor Management API is a simple and efficient REST API designed to manage and retrieve essential information about blood donors. This API helps users quickly access donor details such as name, gender, availability status, blood group code, and a unique donor ID. Using this API, hospitals, emergency services, and individuals can easily search for suitable blood donors during critical situations.

The API is tested and accessed through Postman, where users can send requests to add new donors, update existing donor information, or fetch donor details based on specific requirements.

This API makes the process of finding the right blood donor faster, more organized, and more reliable, especially in emergency scenarios where every second matters.

API Overview & Architecture

1. Introduction to the API

The Blood Donor Management API is a RESTful service designed to store, manage, and retrieve essential information about blood donors. It provides a structured way for applications and users to interact with donor data through simple HTTP requests. The API handles important donor details such as name, gender, availability status, blood group code, and a unique donor ID.

This API allows clients to perform key operations like adding new donors, updating existing donor information, viewing donor lists, searching donors, and deleting donor records. All interactions are done securely through the API without directly accessing the database.

By using Postman for testing, users can easily send requests and receive responses in a clear JSON format, making the development and verification process smooth and efficient. This API ensures that the process of finding a suitable blood donor becomes faster, more reliable, and highly accessible—especially during emergency situations.

2. Base URL

The Blood Donor Management API is accessible using the following base URL:

https://692abcdd7615a15ff24d8666.mockapi.io/donors

3. Main API Endpoint

Field	Type	Description	
DonorID	**String**	**Unique identifier to each donor**	
DonorName	**String**	**Full Name of the Blood donor**	
Availability	**Boolean**	**Donor availability ststus**	
BloodGroup	**String**	**Blood group code of the donor**	

4.CRUD Operations Overview

The Blood Donor Management API supports all four essential CRUD operations, allowing users to efficiently handle donor information through the /donors endpoint. These operations help in adding, retrieving, updating, and deleting donor records in a structured and reliable way.

1. Create (POST)

Creates a new donor record by adding details such as DonorName, Gender, Availability, and BloodGroup. A unique DonorID is generated for each newly added donor.

2. Read (GET)

Retrieves donor information from the system.

This can be used to:

View all donors

Fetch a specific donor using DonorID

3. Update (PUT)

Updates the existing donor's information.

Users can modify details such as availability status, name, gender, or blood group based on requirements.

4. Delete (DELETE)

Removes a donor record from the database using their DonorID.

This ensures outdated or unnecessary donor entries can be safely deleted.

5. Environment Variables Used

In this project, Postman environment variables are used to make API testing easier and more organized. These variables allow the user to change important values in one place without modifying each request manually.

Variable	Description
baseURL	Main API URL
Donor_iD	Holds the DonorID used
Prot	Used only if authentication is added later

6.API Architecture

The Blood Donor Management API follows a simple and modular REST architecture that ensures easy data handling and smooth communication between the client and the server. The architecture is designed to be lightweight, scalable, and easy to test using tools like Postman.

1. Client Layer (Postman / Application)

This layer sends HTTP requests to the API.

Clients can perform operations such as adding a donor, fetching donor data, updating donor details, or deleting a donor using different HTTP methods (GET, POST, PUT, DELETE).

2. API Layer (Endpoints & Routing)

All incoming requests are routed through the /donors endpoint.

The API layer identifies the type of request and maps it to the corresponding operation:

POST → Create a new donor

GET → Read donor information

PUT → Update existing donor records

DELETE → Remove a donor using DonorID

3. Logic Layer (Validation & Processing)

Before storing or returning data, the API performs:

Data validation (e.g., checking if blood group is valid)

Processing logic (e.g., converting Boolean availability values)

Generating unique DonorID .This layer ensures the data remains clean, consistent, and correct.

4. Data Layer (Database / JSON Storage)

This layer stores all donor information, including:

DonorID

DonorName

Gender

Availability status

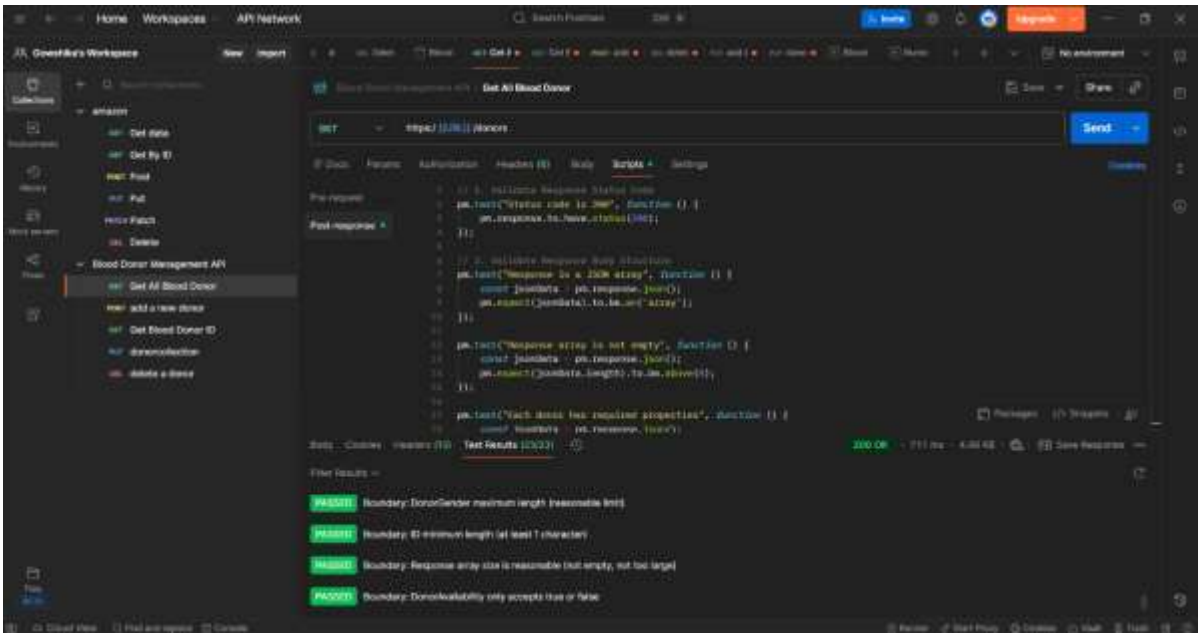
Blood group

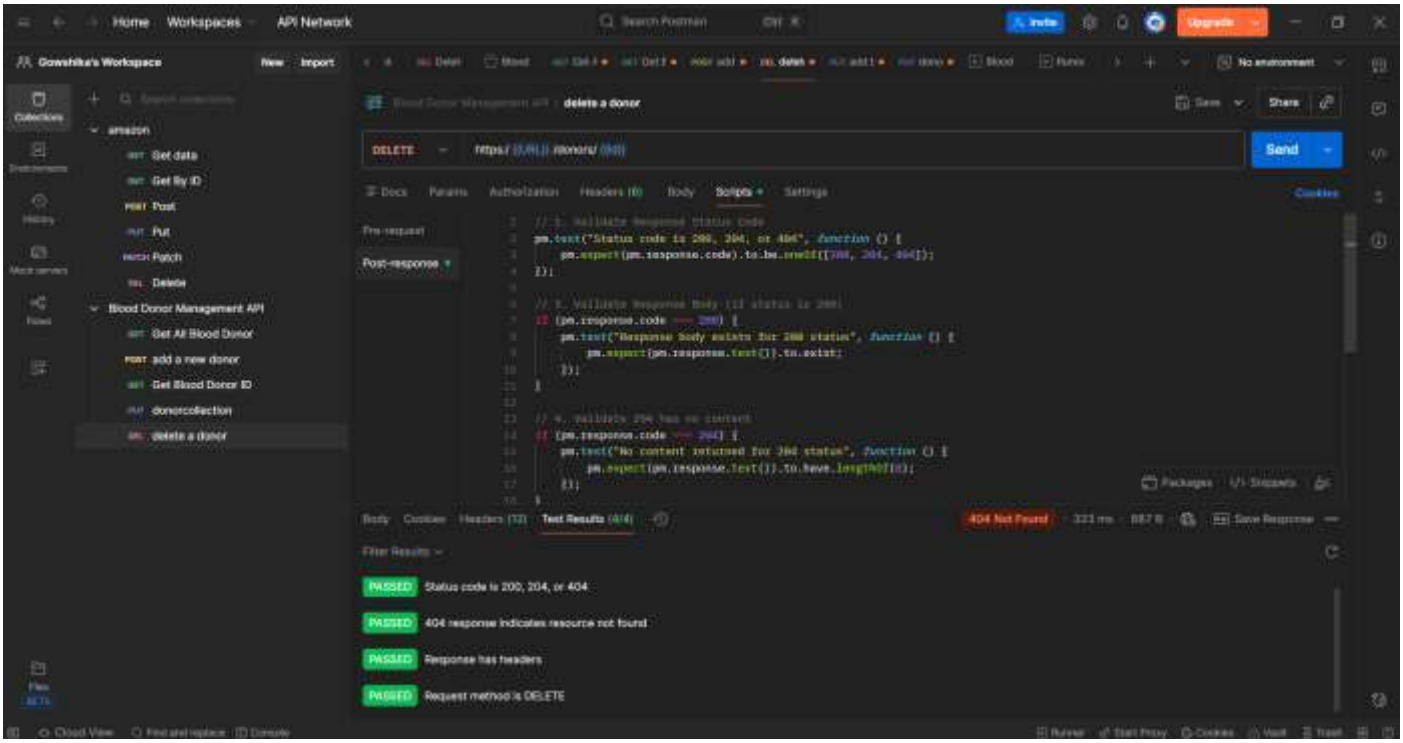
The API interacts with this layer to fetch or update donor records whenever a request is made.

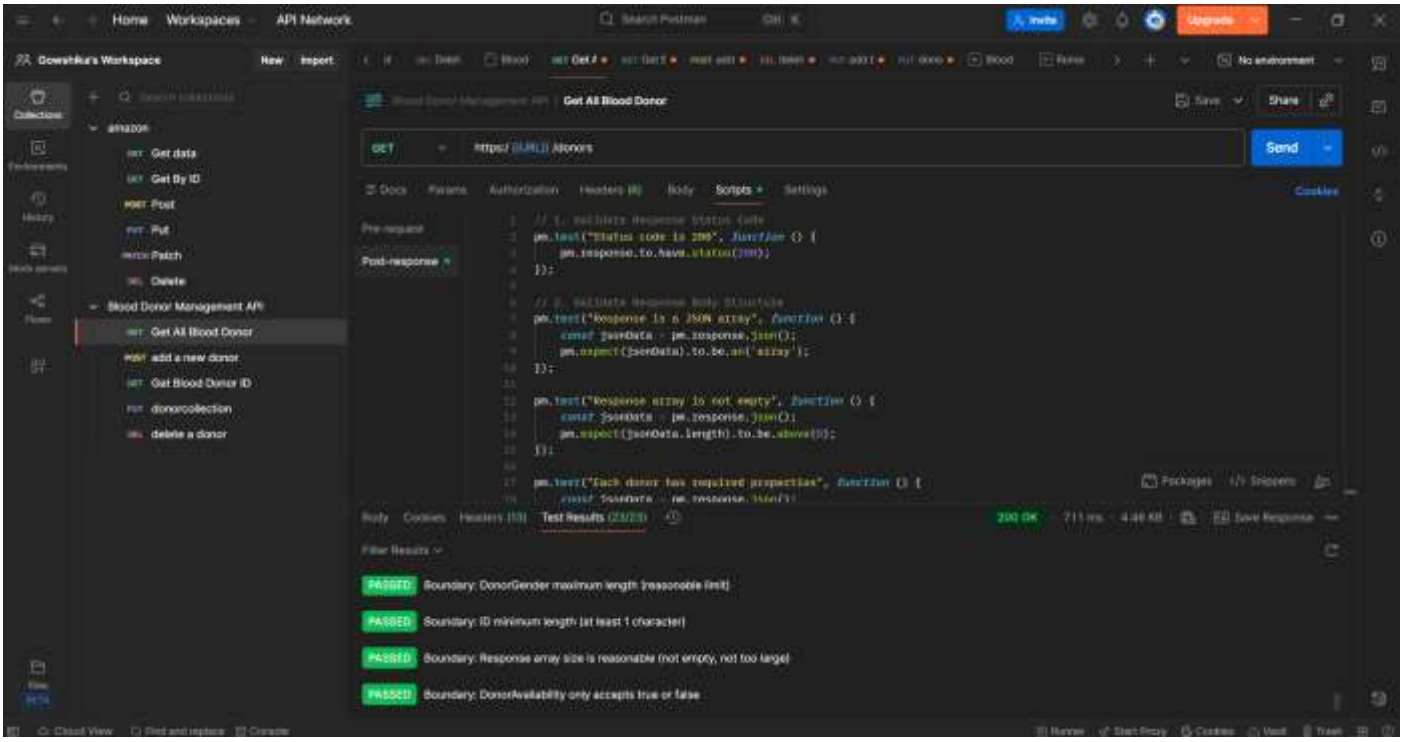
7. Newman and HTML Extra Reporter

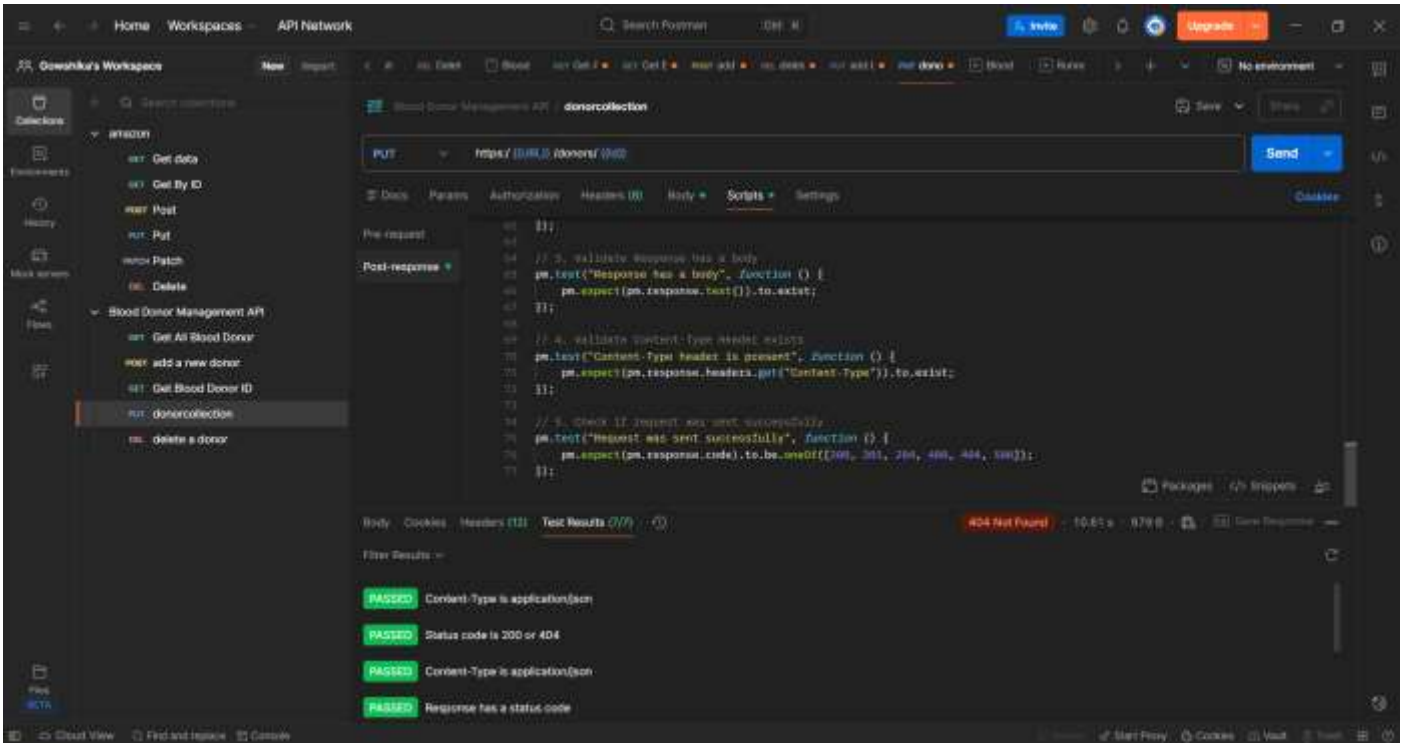
Newman is the command-line tool for running Postman collections without using the Postman application. It enables automated execution of API tests directly through the terminal, making it ideal for CI/CD pipelines and repeated testing workflows. Developers can run collections, apply environments, and export results with simple commands like:

9.ScreenShots




Home
Workspaces
API Network
Gowshika's Workspace
New
Import
amazon
Get data
Get By ID
Post
Put
Patch
Delete
Blood Donor Management API
Get All Blood Donor
add a new donor
Get Blood Donor ID
donorcollection
delete a donor
delete a donor
DELETE
Send
Docs
Params
Authorization
Headers (8)
Body
Scripts
Settings
Pre-request
Post-response
404 Not Found
Test Results
PASSED
Status code is 200, 204, or 404
PASSED
404 response indicates resource not found
PASSED
Response has headers
PASSED
Request method is DELETE


Home
Workspaces
API Network
Gowshika's Workspace
New
Import
amazon
Get data
Get By ID
Post
Put
Patch
Delete
Blood Donor Management API
Get All Blood Donor
add a new donor
Get Blood Donor ID
donorcollection
delete a donor
Get All Blood Donor
GET
Send
Docs
Params
Authorization
Headers (8)
Body
Scripts
Settings
Pre-request
Post-response
200 OK
Test Results
PASSED
Boundary: DonorGender maximum length (reasonable limit)
PASSED
Boundary: ID minimum length (at least 1 character)
PASSED
Boundary: Response array size is reasonable (not empty, not too large)
PASSED
Boundary: DonorAvailability only accepts true or false



10.CONCLUSION

The Blood Donor Management API provides a simple, efficient, and structured solution for managing blood donor information. With support for essential CRUD operations, the API allows users to add, retrieve, update, and delete donor records with ease. Its REST-based design makes it easy to integrate with any client application and ensures smooth communication between the client and the server.

By using Postman for testing, the API becomes easy to validate, monitor, and maintain. The clear endpoint structure and organized data fields help ensure that donor information is accurate, accessible, and updated in real time. This system can be highly useful in emergency situations where quick access to the right blood donor is crucial.

Overall, the Blood Donor Management API is a reliable and user-friendly backend service designed to support faster decision-making, better data handling, and improved coordination in blood donation processes.