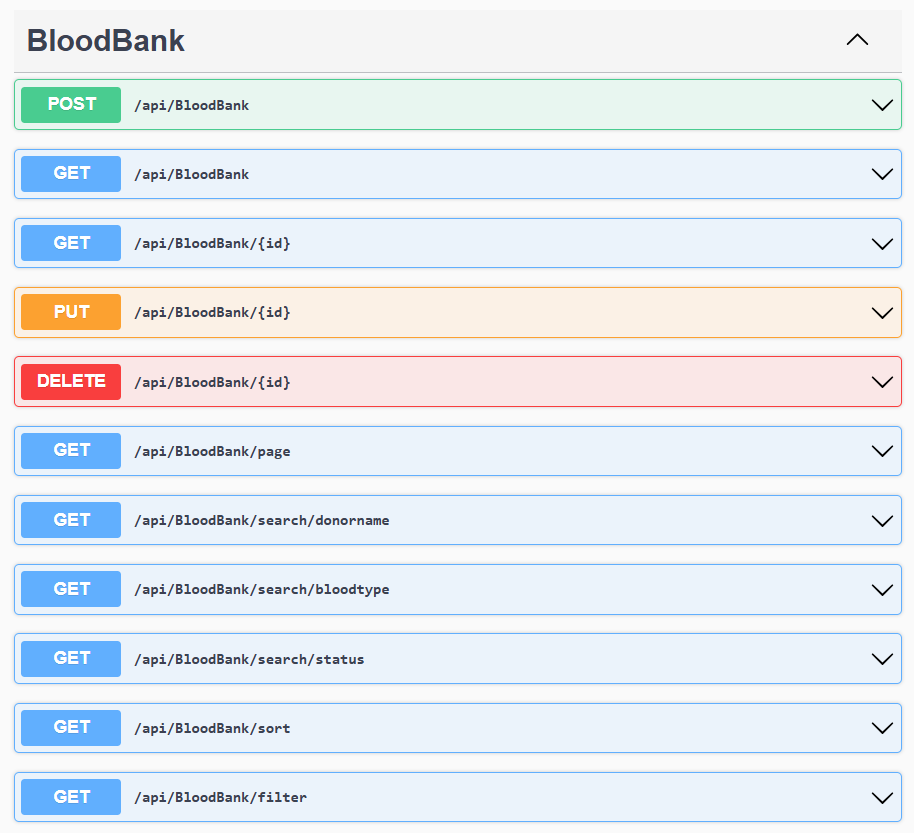
**Blood Bank Management REST API Assignment Explanation Docs** 

**1.Post: POST /api/BloodBank**

This endpoint accepts the creation of a new blood bank entry. Here are the detailed validations that are performed when a request is made to this endpoint:

**Age Validation**

* **Condition**: The donor’s age must be between 18 and 65 years.
* **Error Message**: Age must be between 18 and 65.

**Quantity Validation**

* **Condition**: The quantity of blood must be greater than 0.
* **Error Message**:Quantity should be greater than 0.

**Contact Info Validation**

* **Condition**: The contact information (email) must be a valid email address (i.e., no "@" symbol).
* **Error Message**:ContactInfo must be a valid email.

**Collection Date Validation**

* **Condition**: The collection date should not be in the future.
* **Error Message**: CollectionDate cannot be in the future.

**Expiration Date Validation**

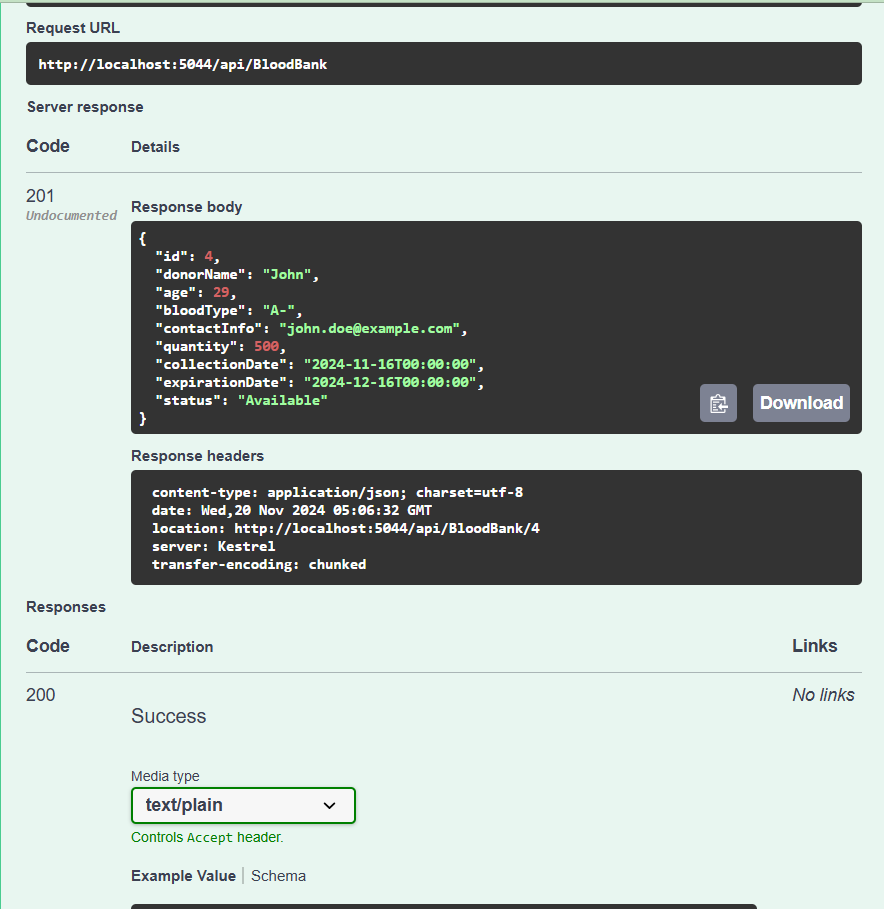
* **Condition**: The expiration date must be after the collection date.
* **Error Message**:ExpirationDate must be after CollectionDate.

**Status Validation**

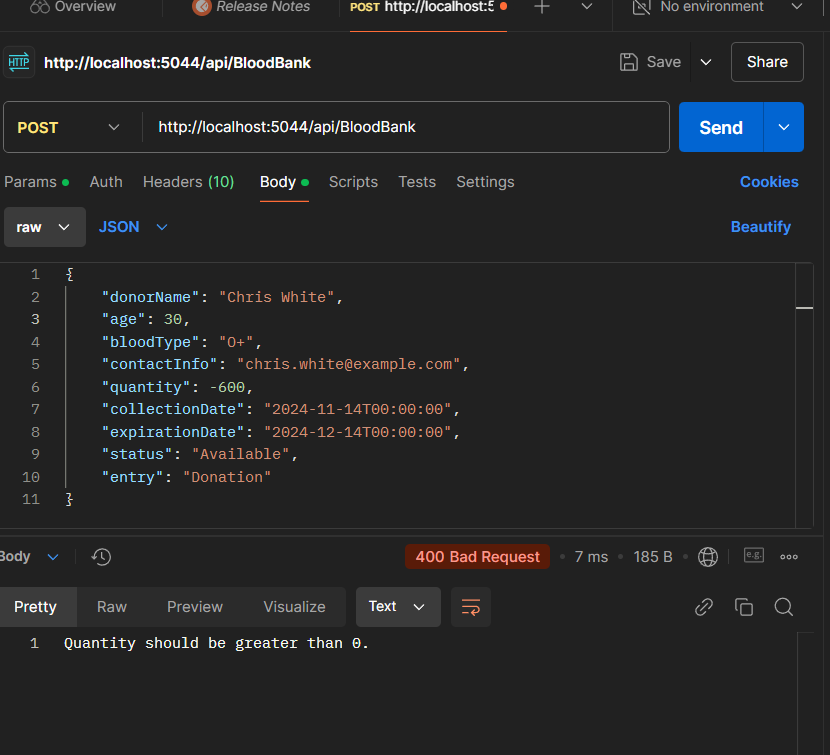
* **Condition**: The status must be one of the following: "Available", "Requested", or "Expired".
* **Error Message**:Status not valid

If any of the above validation conditions are not met, a **Bad Request (400)** response is returned with the specific error messages for each failed validation.

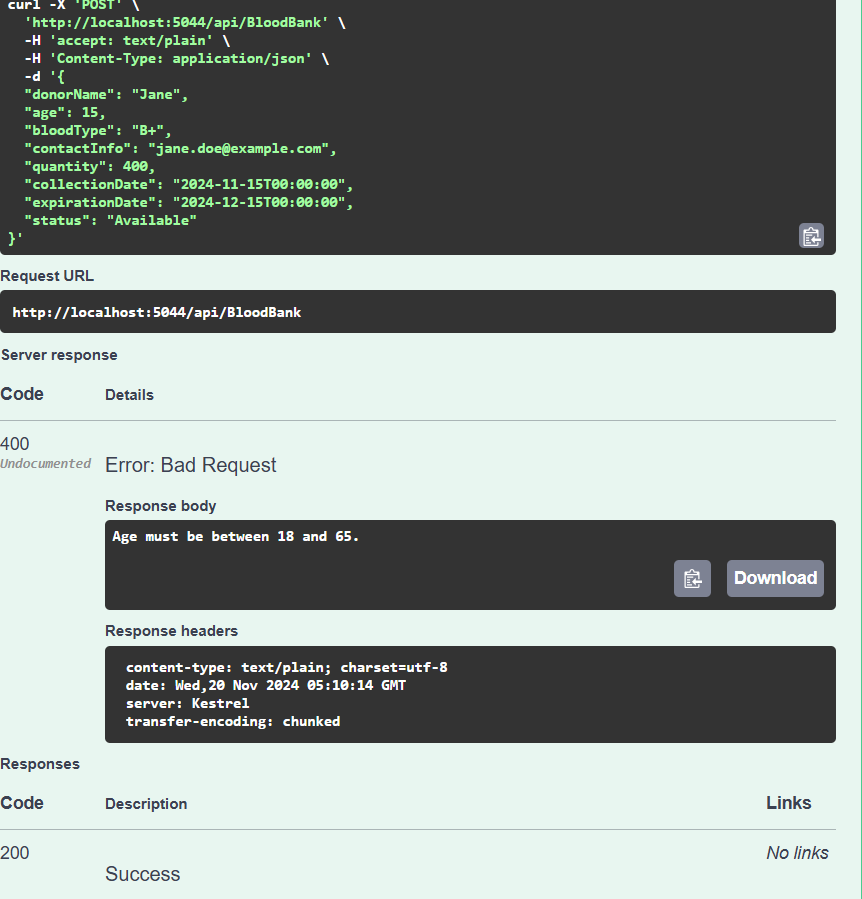
Valid post request:



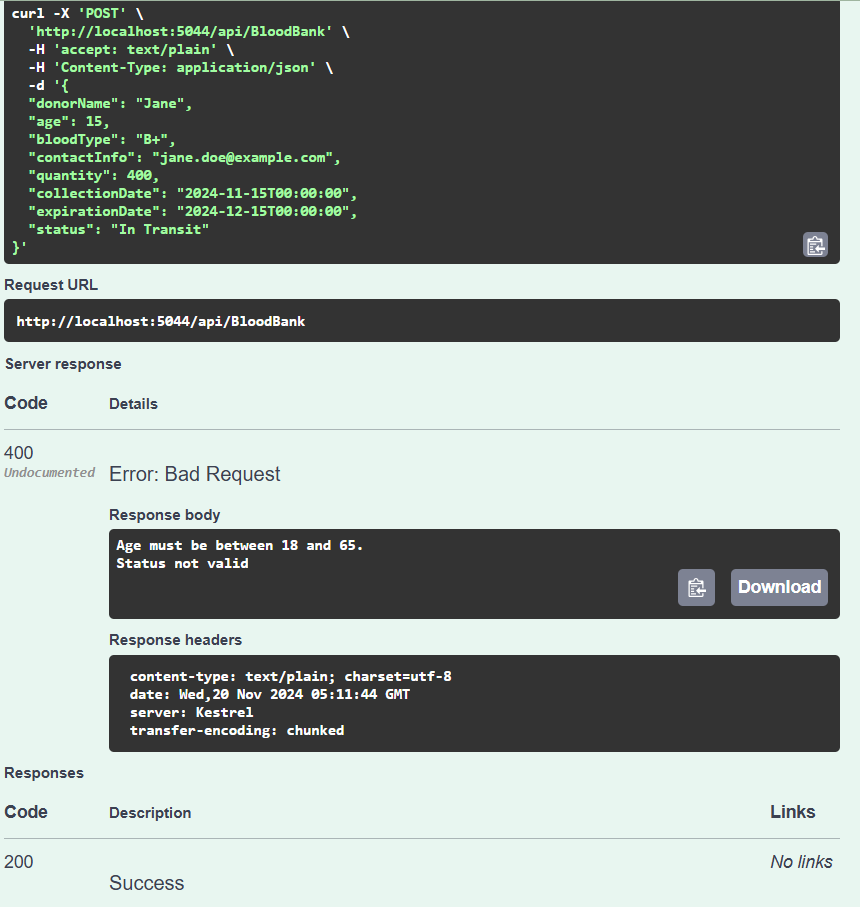
Invalid Quantity:



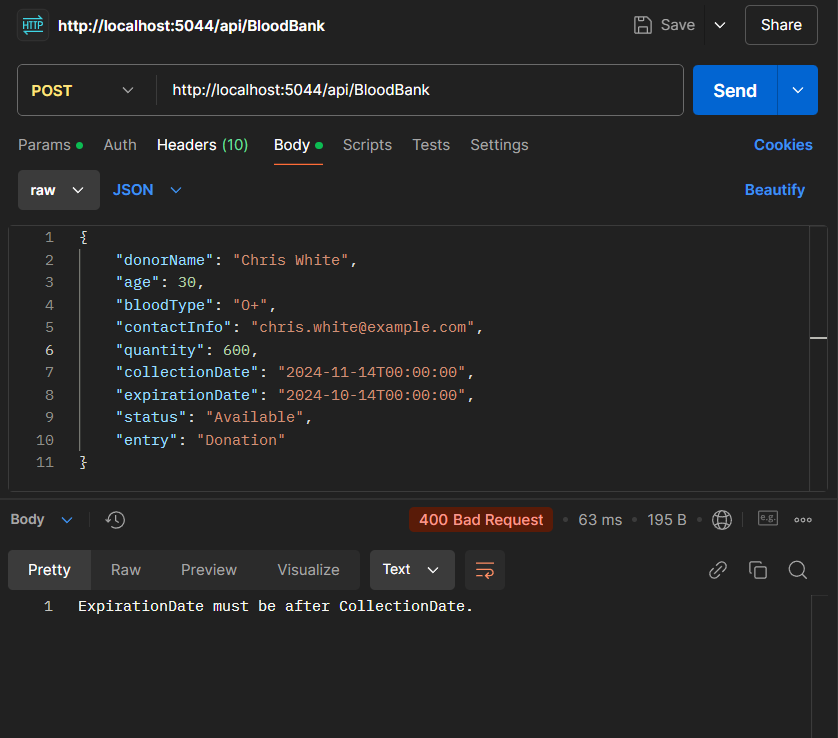
Invalid Age:



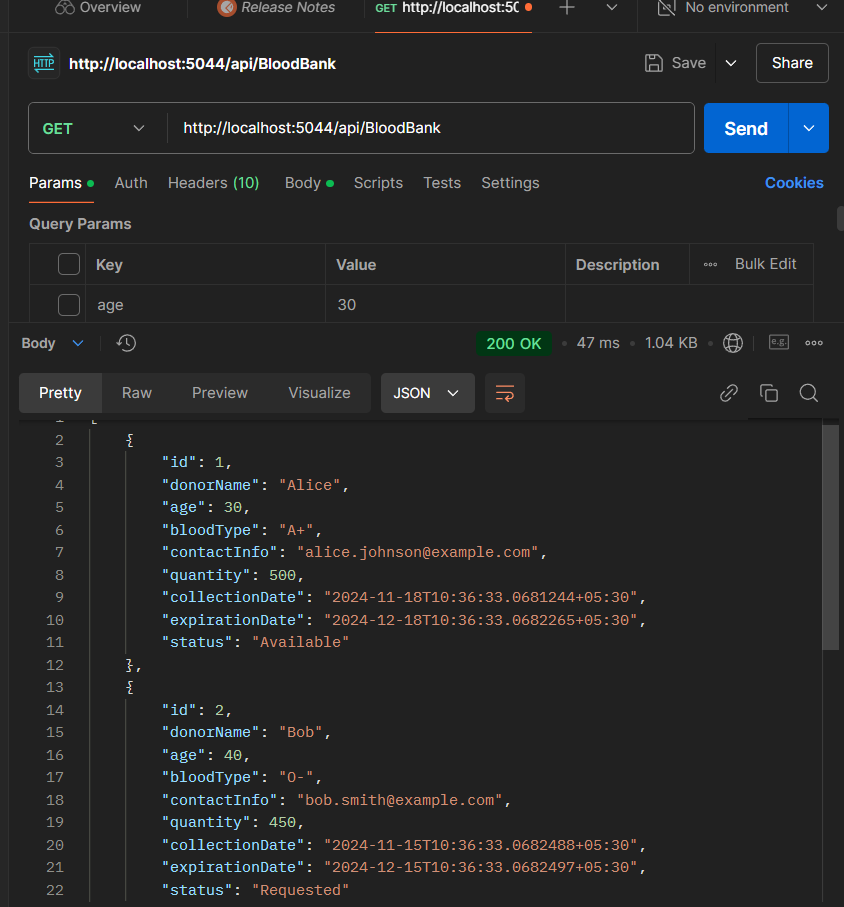
Invalid Status

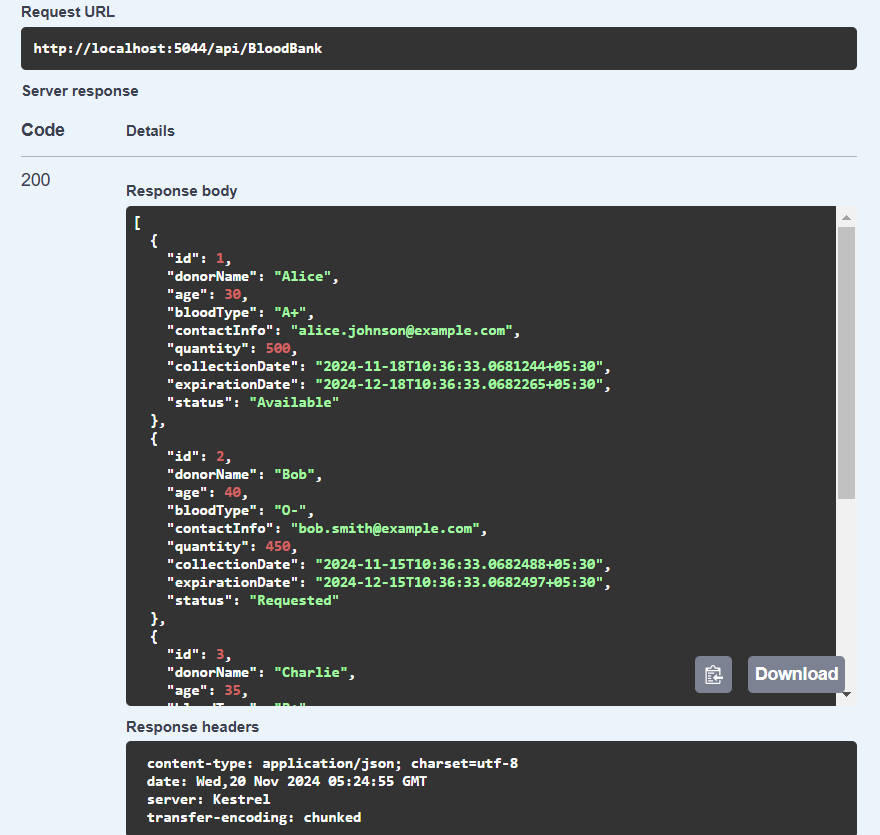


Invalid ExpirationDate

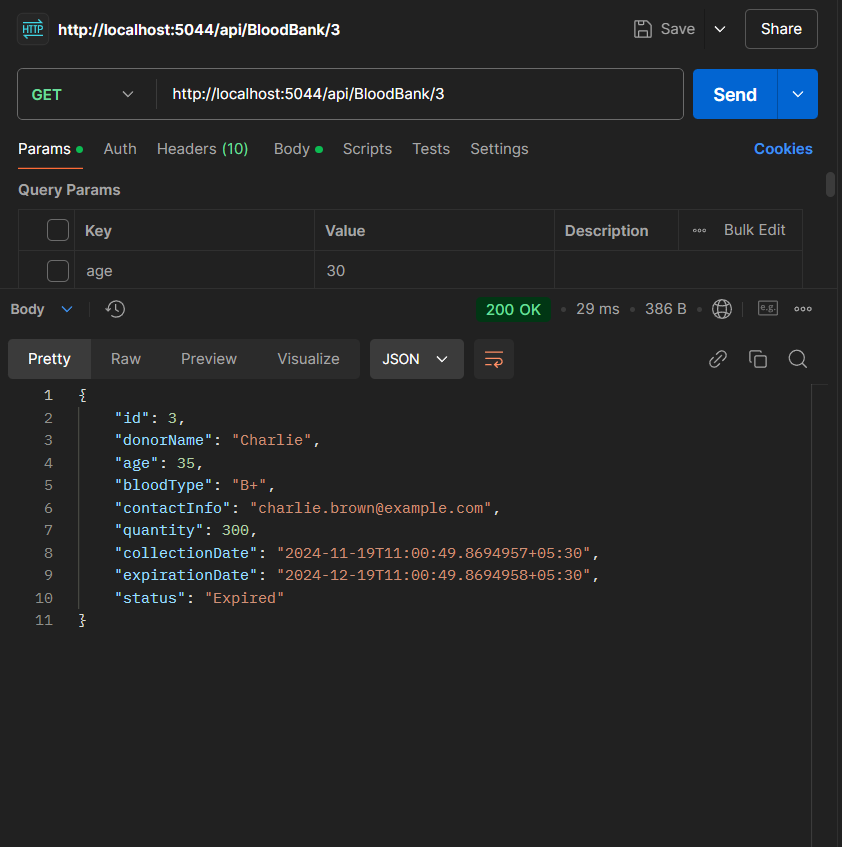


**2.Get: GET /api/BloodBank** – Get all blood bank entries

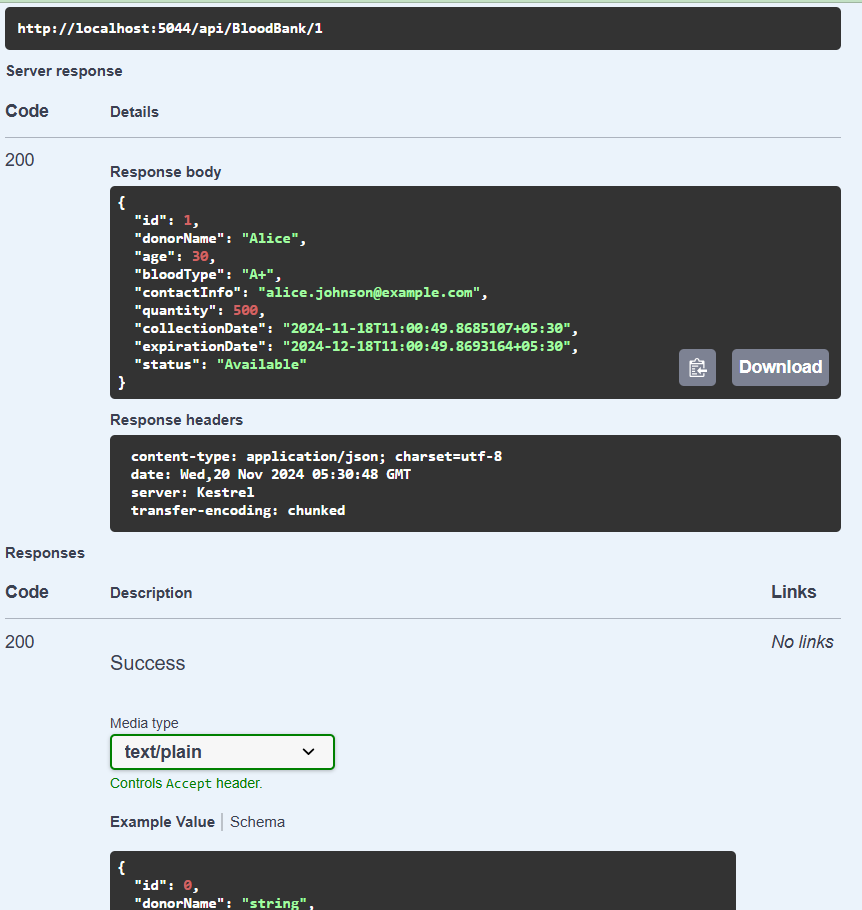




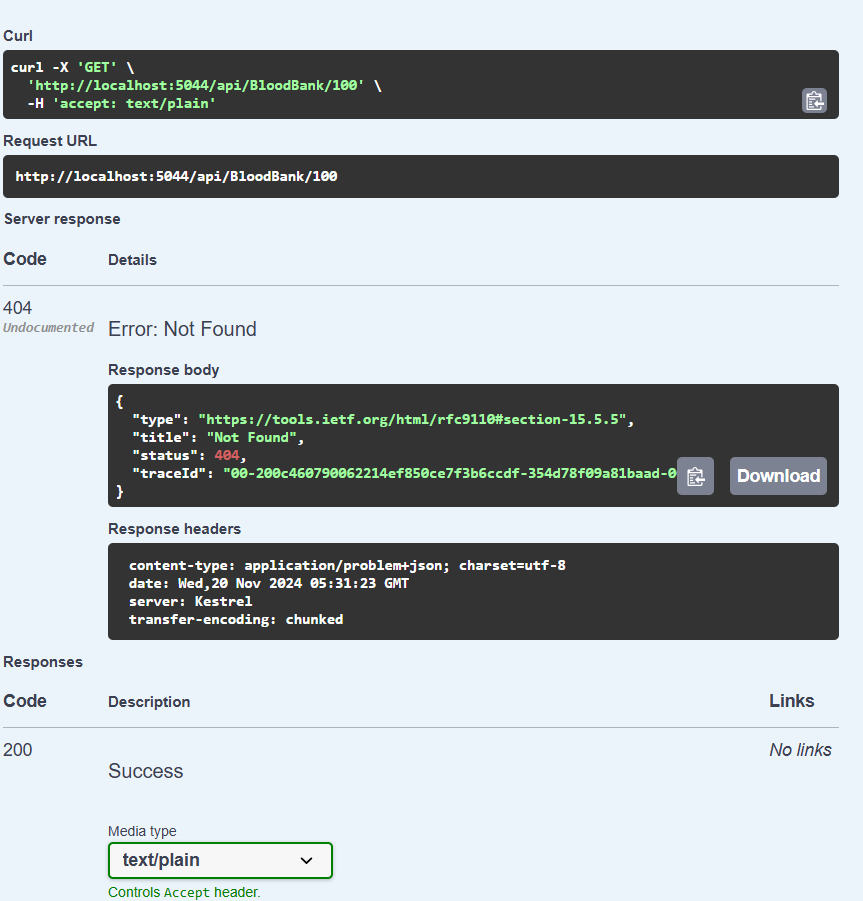
**3. GET /api/BloodBank/{id}** – Get blood bank entry by ID

This test case checks that a valid blood bank entry can be retrieved by specifying its ID in the URL. The system return the corresponding blood bank details if the ID exists. The response include the requested entry with an HTTP status of 200 OK.  
  
Valid: when id=3  


Valid when id=1:

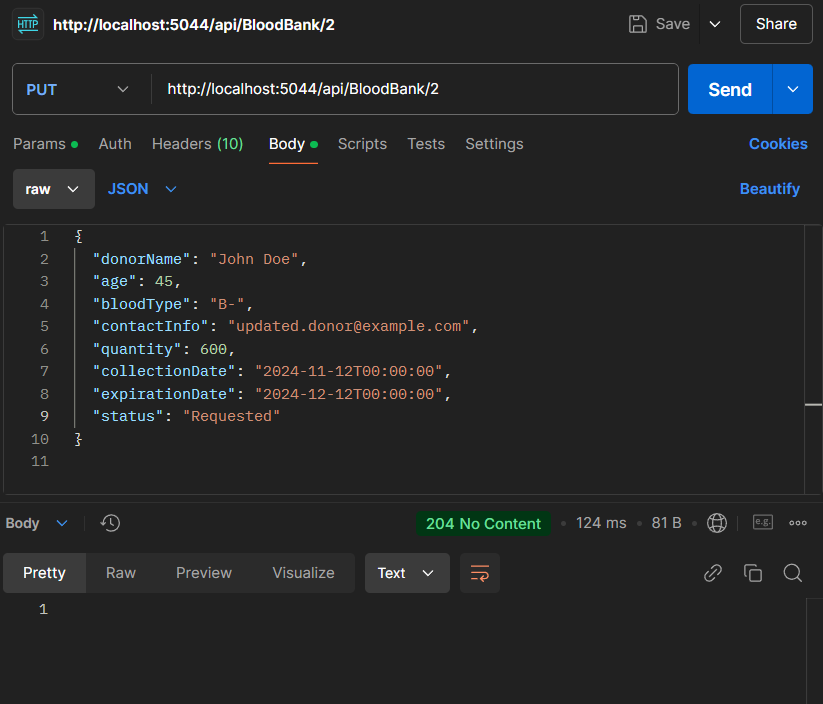


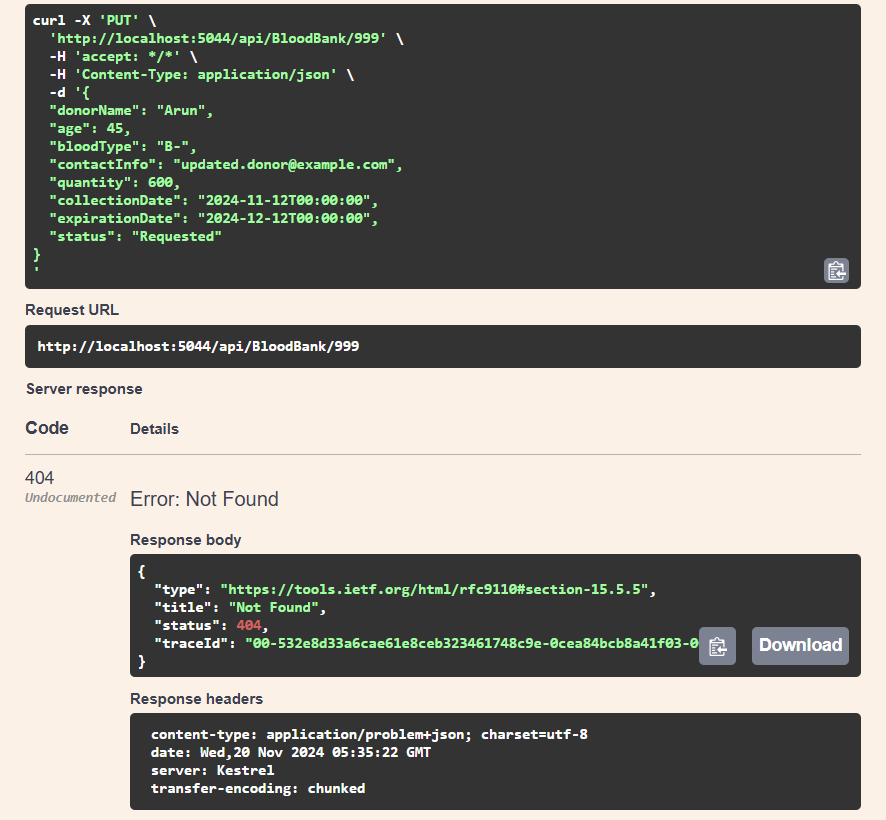
**Invalid: Id =100 does not exist**



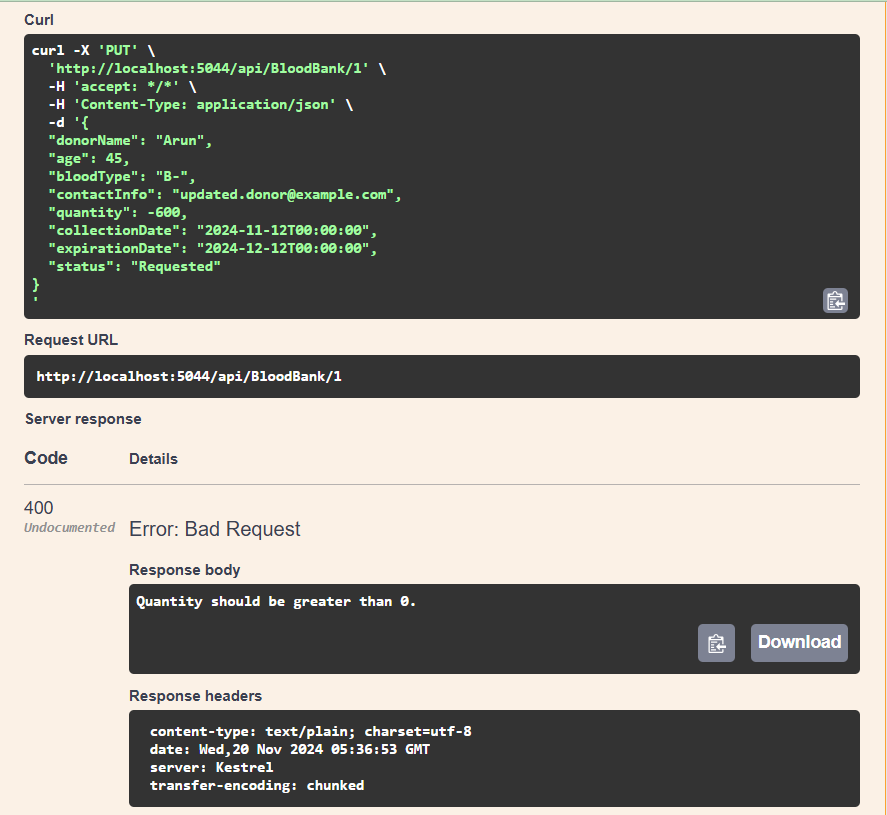
**4. PUT /api/BloodBank/{id}** – Update an existing blood bank entry

**Valid:**This test case tests updating an existing blood bank entry by providing the updated data for the specified ID. The request body should new values for fields such as donor name, quantity, collection date, etc. If the entry is found, the system update it and return an HTTP status of 204 No Content to indicate that the update was successful.



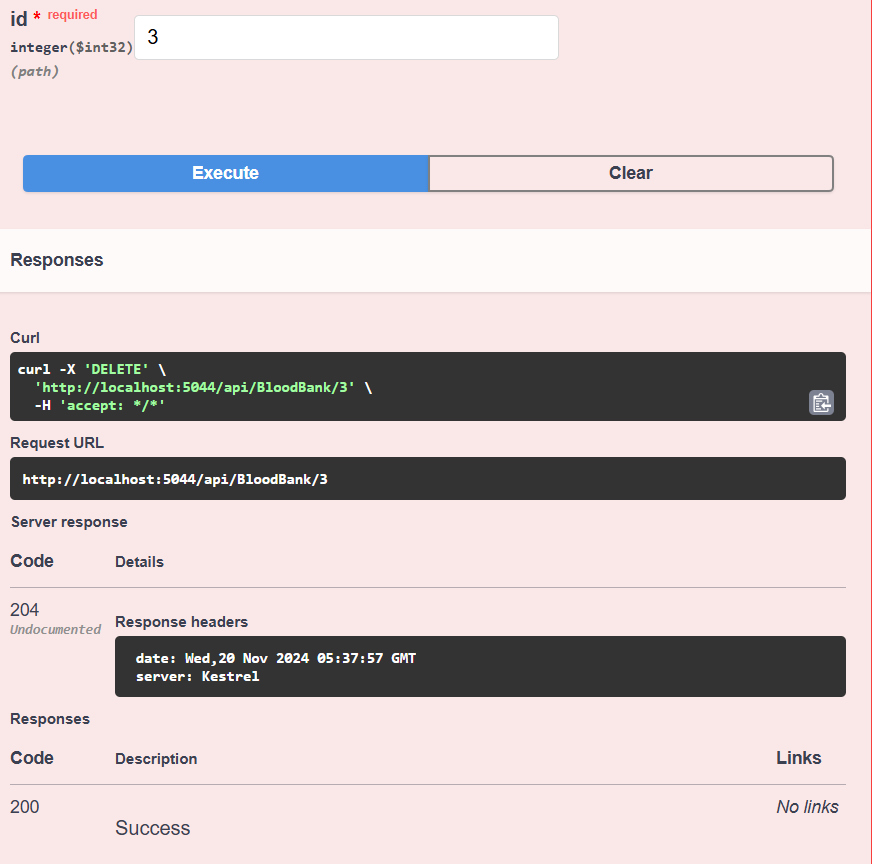
**Invalid:**the specified ID does not exist in the system,so the API return a 404 error with a message indicating that the entry was not found.  


**Invalid:**the updated quantity is negative so the system should reject the request with a 400 error and provide a message.

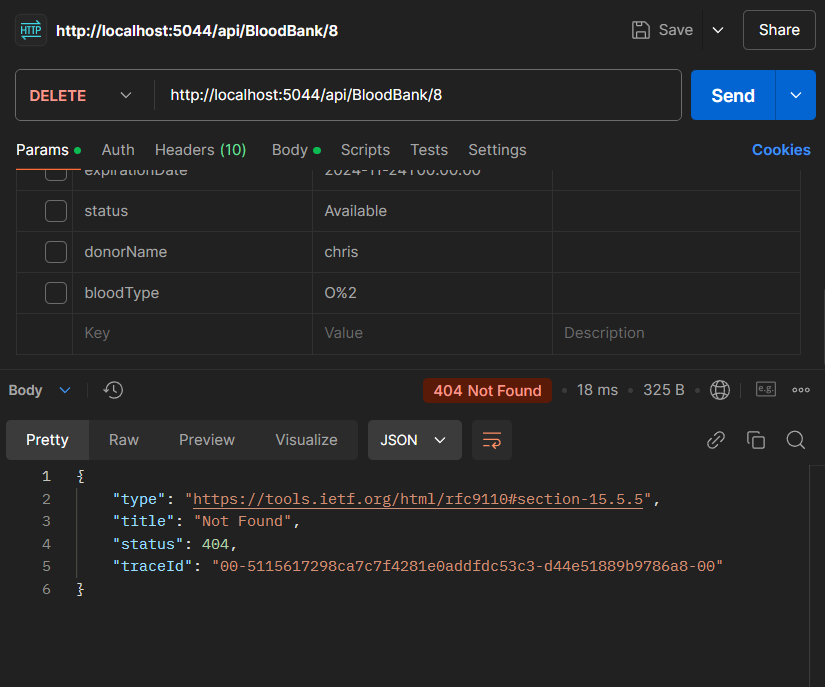


**5. DELETE /api/BloodBank/{id} – Delete a blood bank entry by ID**

**Valid** :This test case verifies that an existing blood bank entry can be deleted using the DELETE method and the entry's ID in the URL. The entry exists, it is removed from the system, and the API should return an HTTP status of 204 No Content to indicate that the deletion was successful.

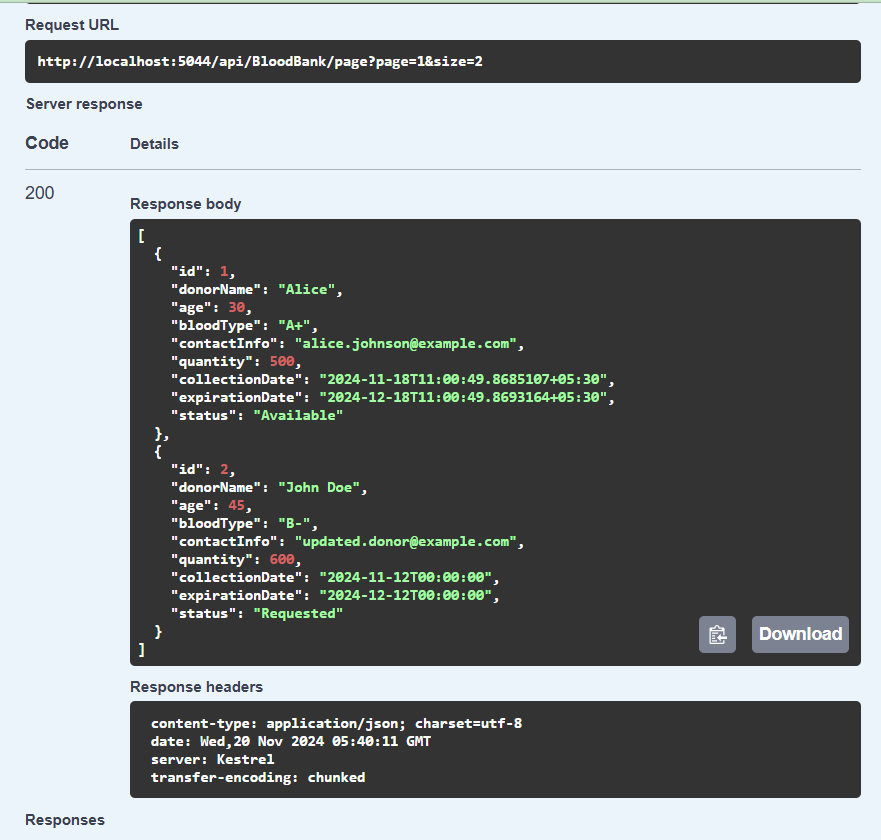


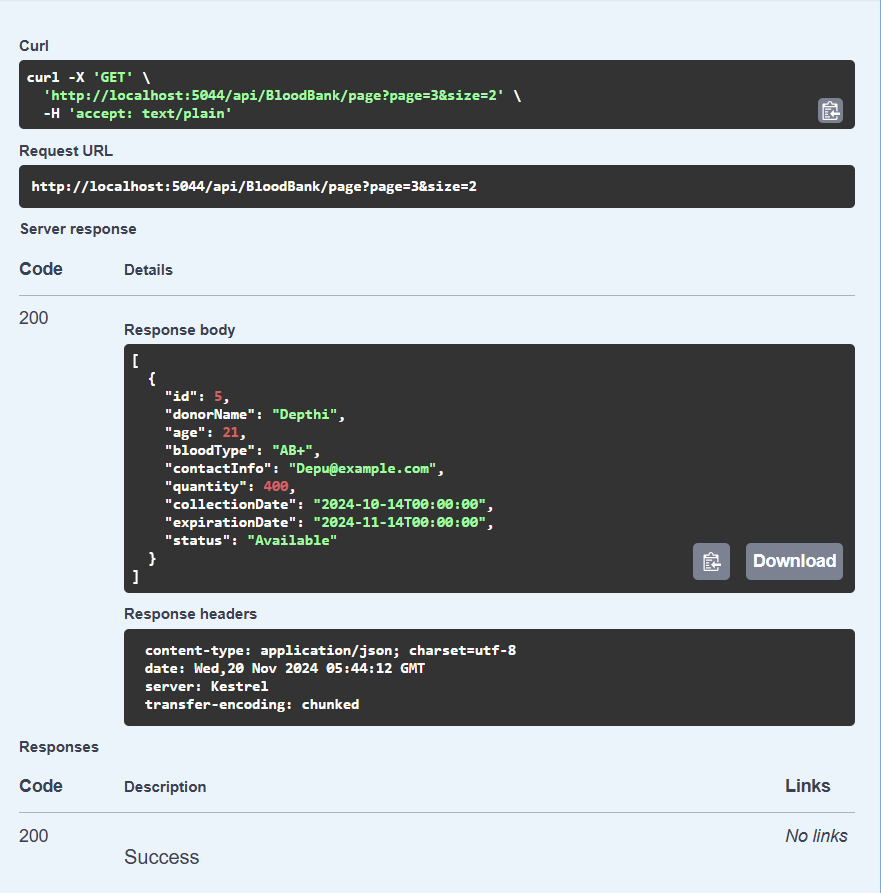
**Invalid**: The specified ID does not correspond to an existing entry in the system, the API return an HTTP status of 404 Not Found, indicating that the entry could not be deleted because it does not exist.



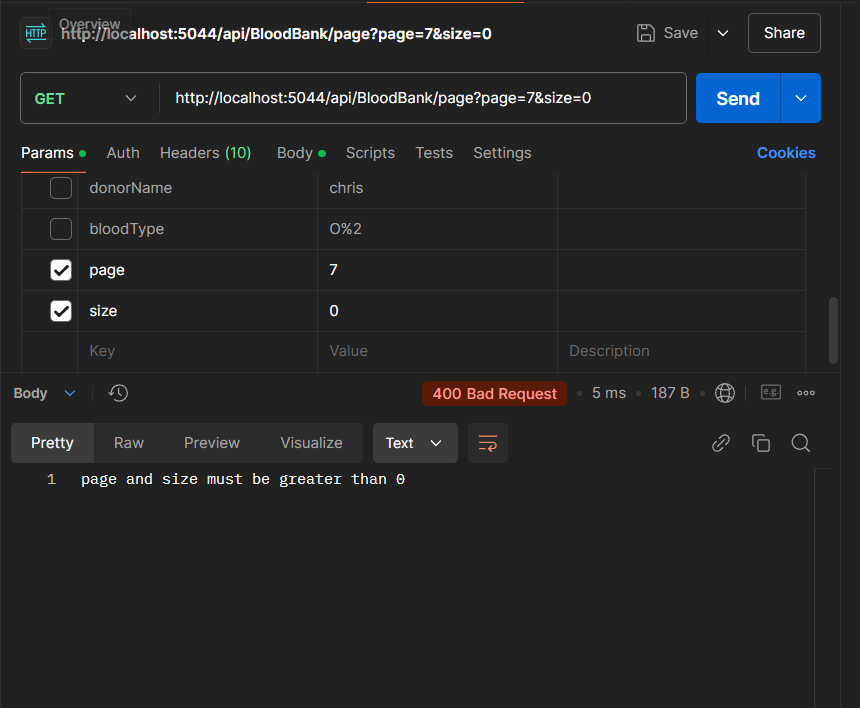
**6. GET /api/BloodBank/page** – Get paginated blood bank entries

**Valid Test Case:** This test case checks that the API supports pagination. By specifying a page number and size, you can retrieve a subset of blood bank entries. The response return the corresponding page of blood banks, with an HTTP status of 200 OK.



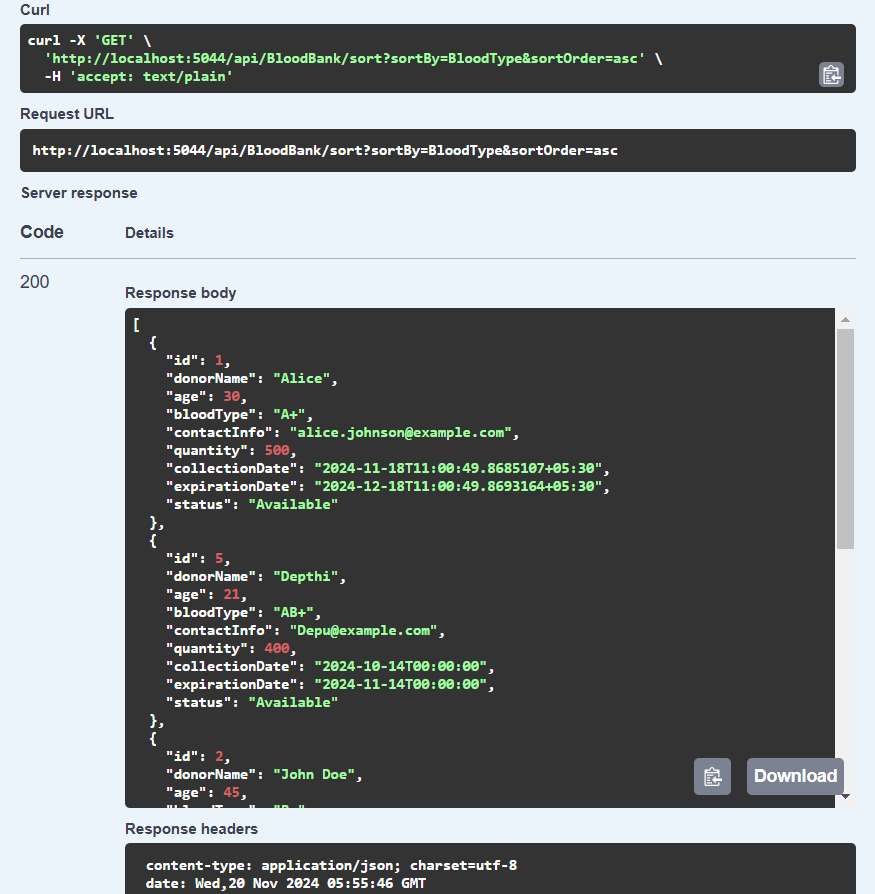


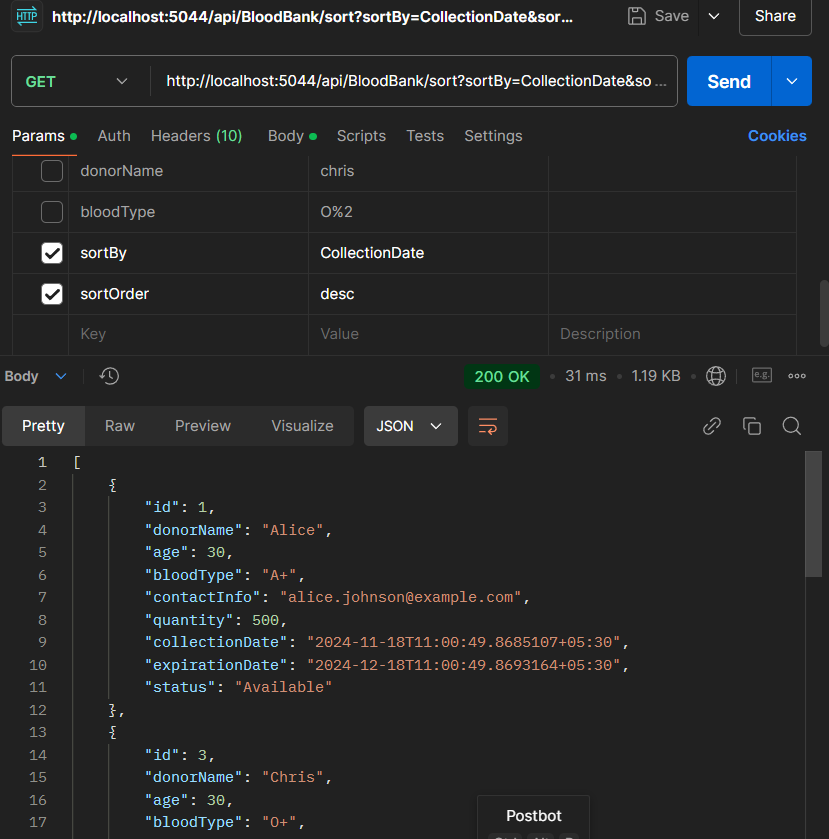
**Invalid :**The page and size parameters must be positive integers. If the page or size is negative or zero, the system should return a 400 error with a message indicating that the page and size must be greater than zero.



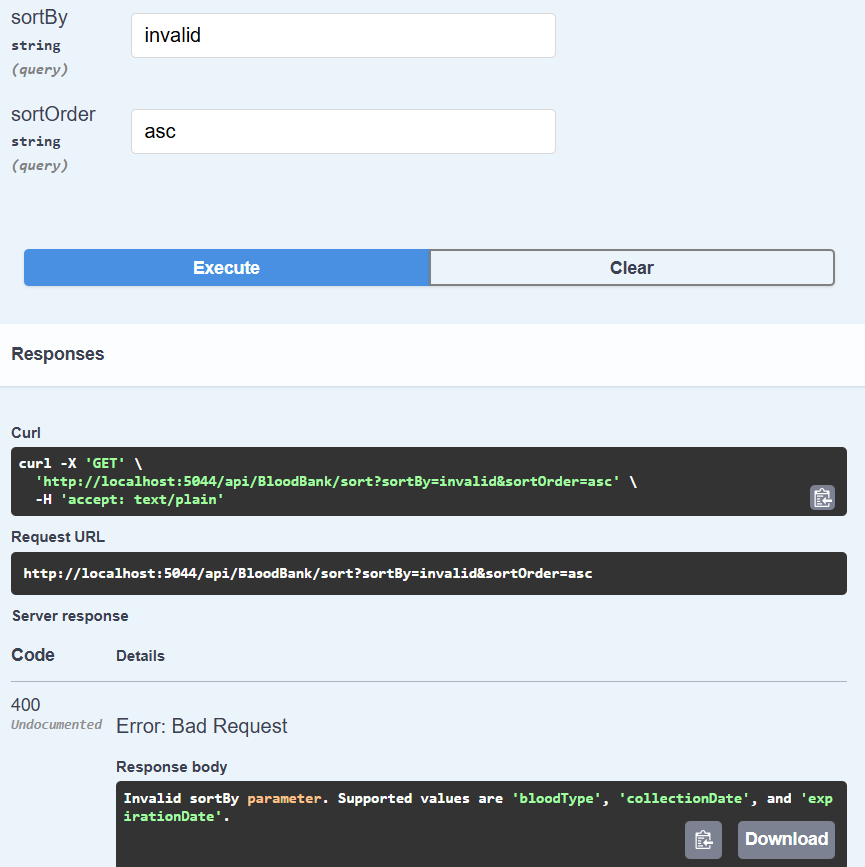
7.**GET /api/BloodBank/sort**– Get sorted blood bank entries

**Valid Test Case:** This test case checks that the API supports pagination. By specifying a sortOrder and Sort By, we can retrieve a blood bank entries in sorted order. The response return the blood banks, with an HTTP status of 200 OK.



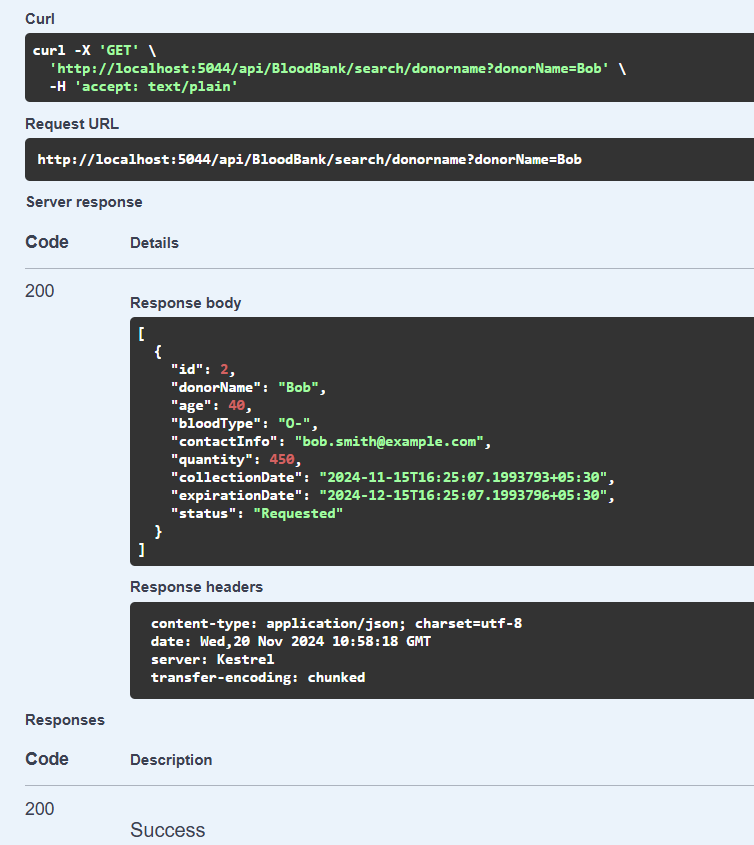
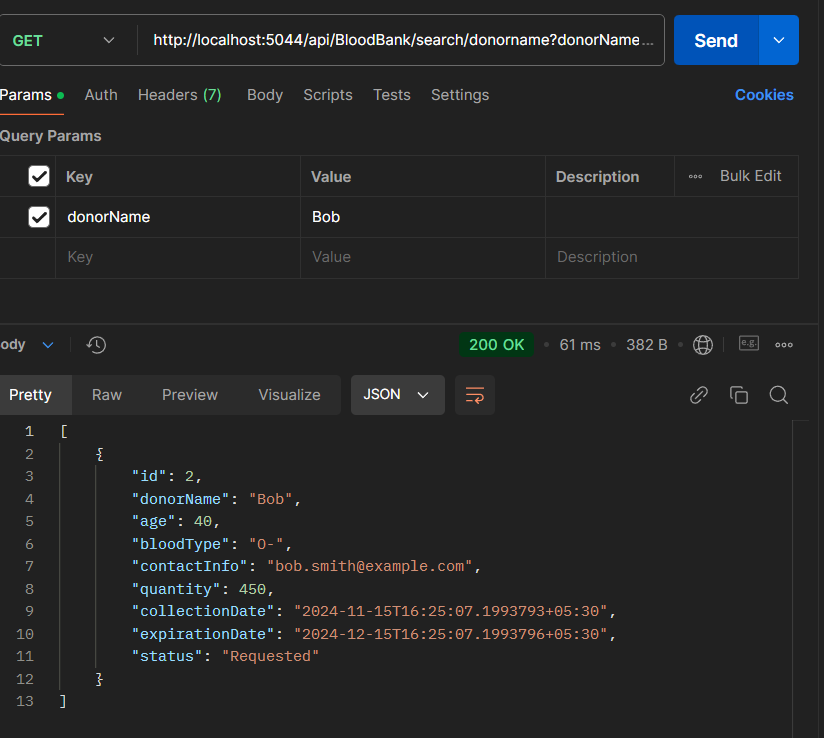


**Invalid :**The sortBy parametersis not valid,so the system should return a 400 error with a message indicating that sortBy parameter is not valid



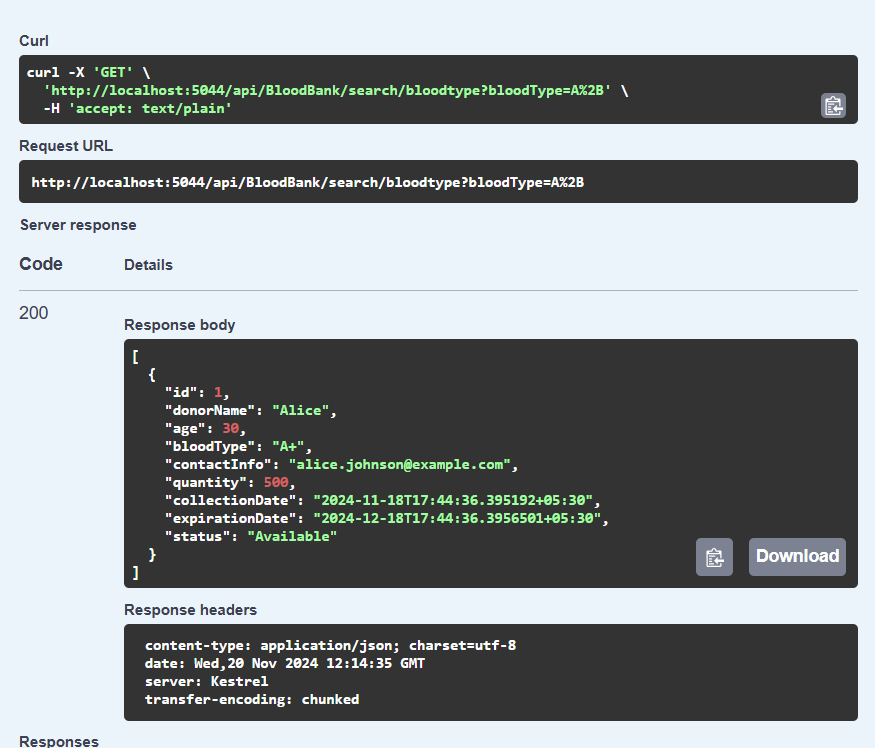
8.Get: **GET /api /bloodbank/search/donorname**:

This endpoint allows users to search for blood bank entries based on the **donor's name**. It performs a **case-insensitive partial match** to find entries where the donor's name contains the given search string.

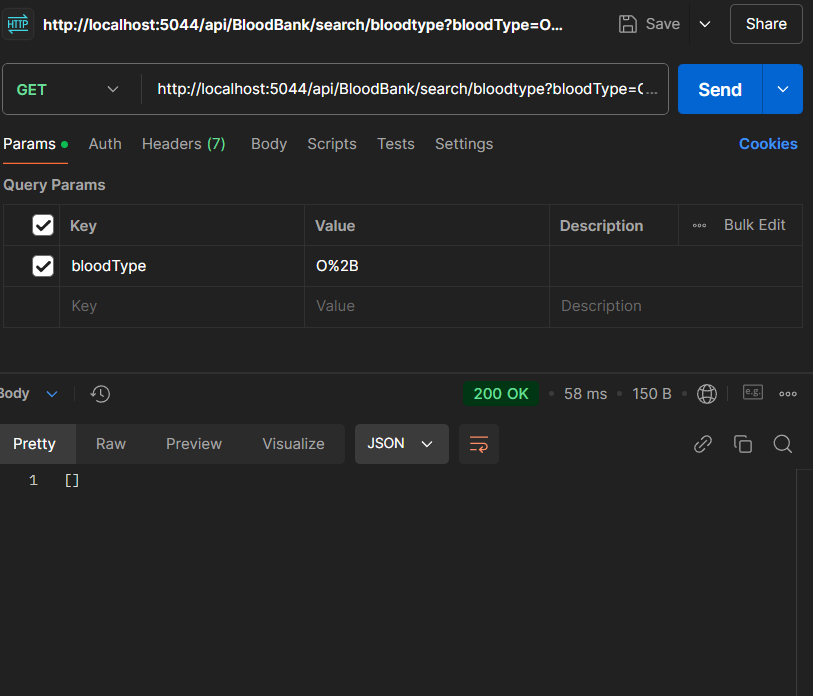
  


9. **GET /api /bloodbank/search /bloodtype**

This endpoint allows users to search for blood bank entries based on the **blood type** (e.g., A+, O-, B+). The search is **case-insensitive** matches.

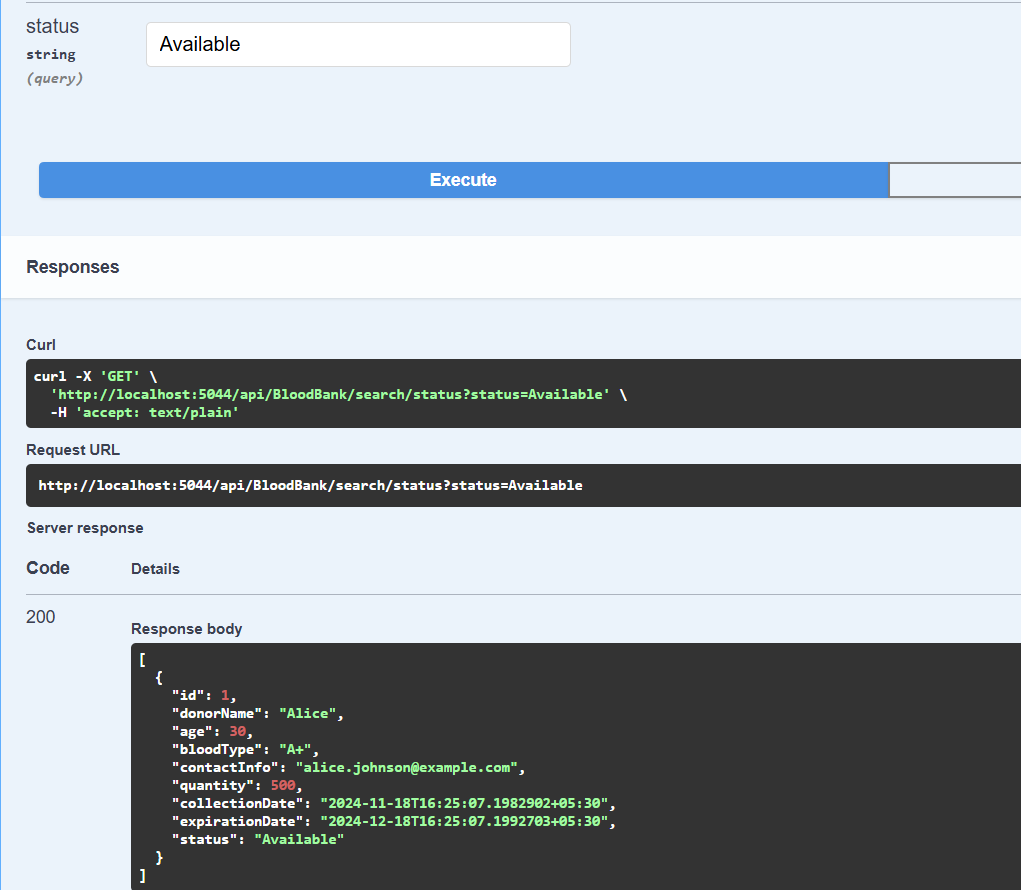


If entries does not match it gives empty list.

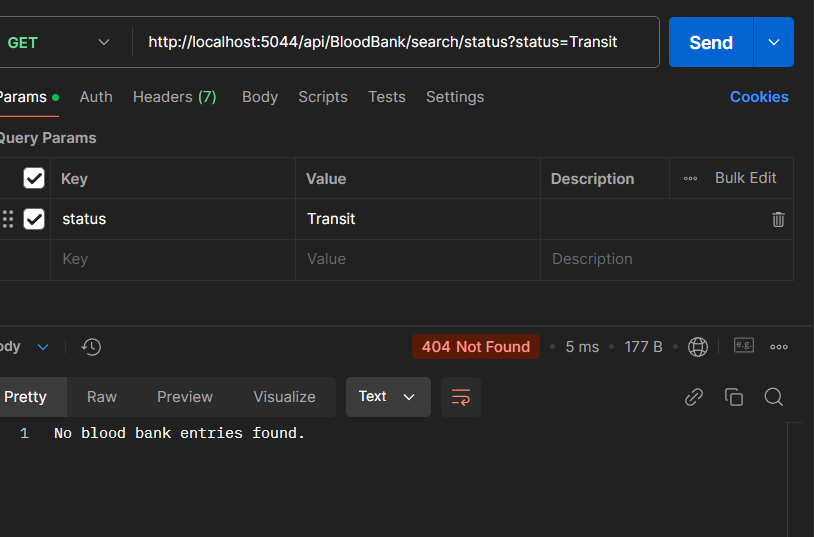


**10. GET /api /bloodbank/search/donorname /status**

This endpoint allows users to search for blood bank entries based on their **status** (e.g., Available, Requested, Expired). The search is **case-insensitive** and supports partial matches.



If there is no match it return 404 with message.



11. **GET /api/BloodBank/filter:** This endpoint allows Allow multiple search parameters to be used simultaneously. The search is **case-insensitive** matches

