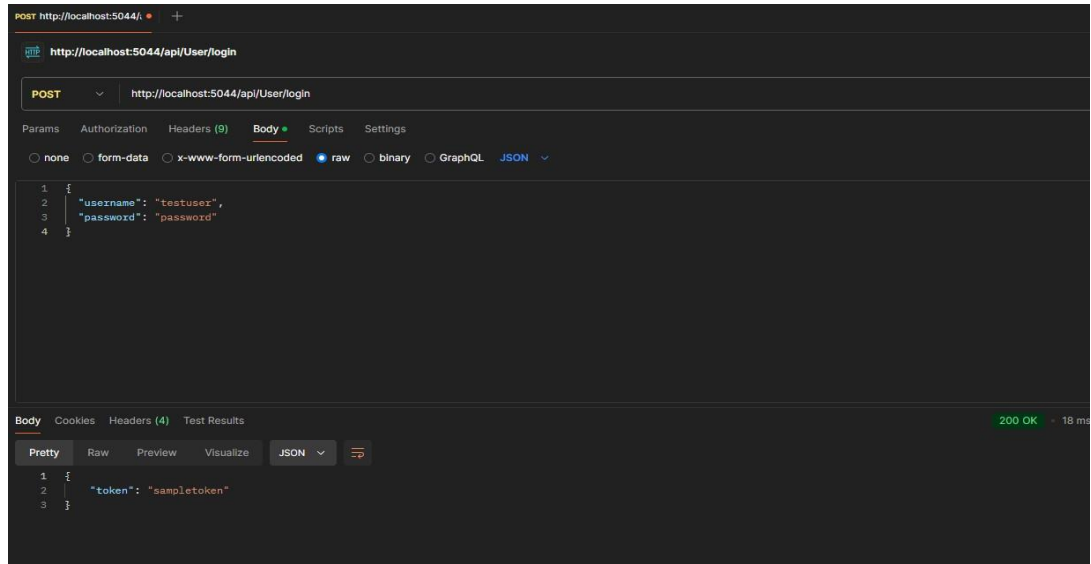


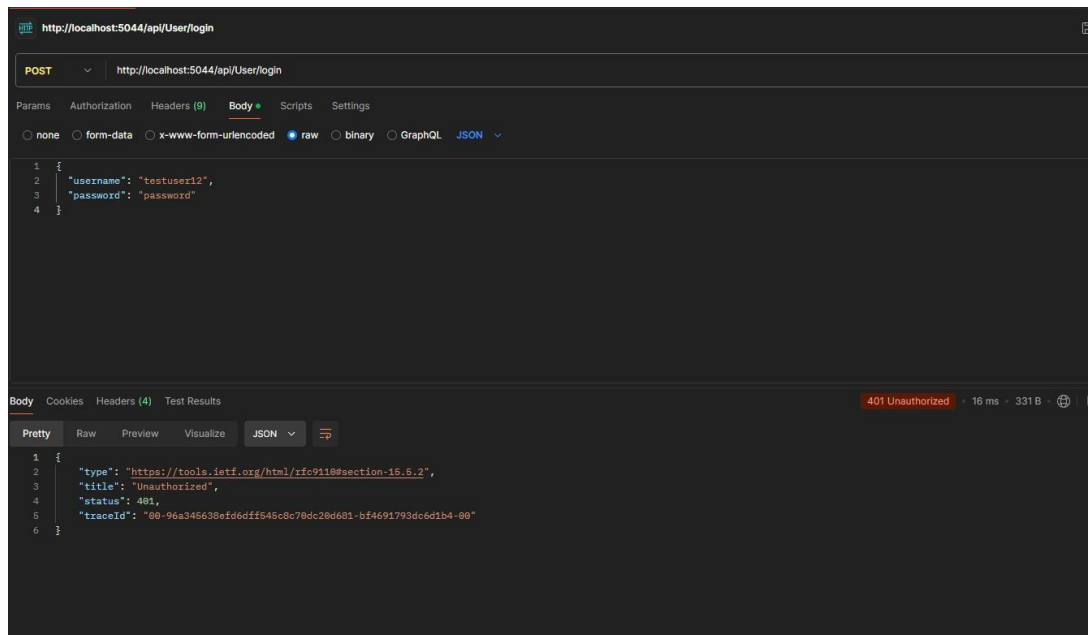
TEST EXECUTION AND REPORTING

Test Execution:

1. AB-01- User login



2. AB-02 User login incorrect



3. AB-03-User Login Usurname Empty

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:5044/api/User/login`
- Method:** `POST`
- Body:** A JSON object with an empty string for the username:

```
1 {
2   "username": "",
3   "password": "password"
4 }
```
- Response:** A `401 Unauthorized` status with a response time of 12 ms and a body size of 331 B. The response body is a JSON object:

```
1 {
2   "type": "https://tools.ietf.org/html/rfc9118#section-15.5.2",
3   "title": "Unauthorized",
4   "status": 401,
5   "traceId": "00-87ba7b5129e164d2ee9f888ef62ce38f-38dc4a511e4d3af2-00"
6 }
```

4. AB-04-User Login password Empty

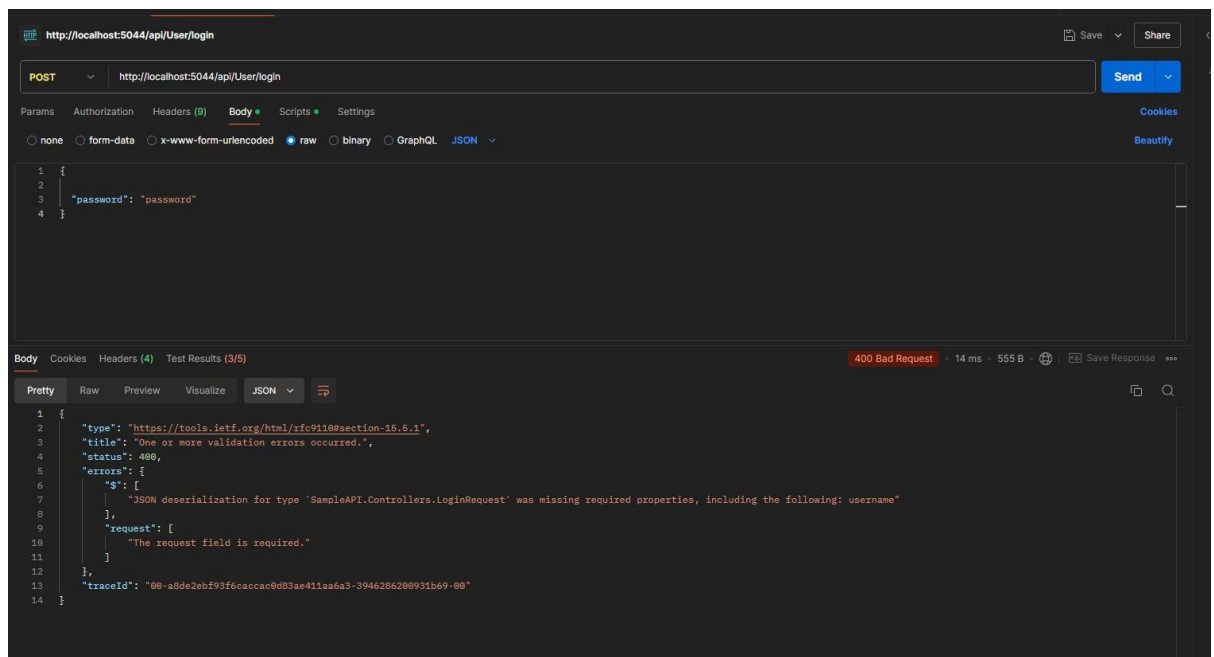
The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:5044/api/User/login`
- Method:** `POST`
- Body:** A JSON object with a valid username and an empty password:

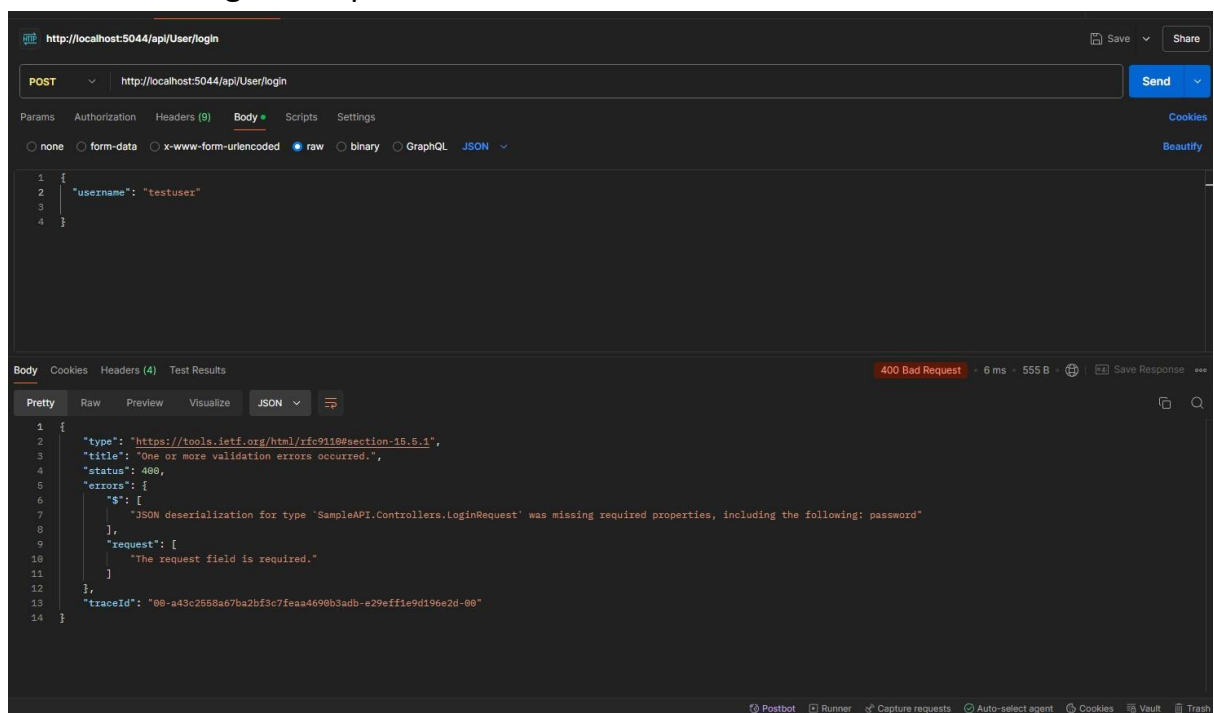
```
1 {
2   "username": "testuser",
3   "password": ""
4 }
```
- Response:** A `401 Unauthorized` status with a response time of 11 ms and a body size of 331 B. The response body is a JSON object:

```
1 {
2   "type": "https://tools.ietf.org/html/rfc9118#section-15.5.2",
3   "title": "Unauthorized",
4   "status": 401,
5   "traceId": "00-d399651c17383b1dd9e7774e02f86d48-ce14adf4abf97544-00"
6 }
```

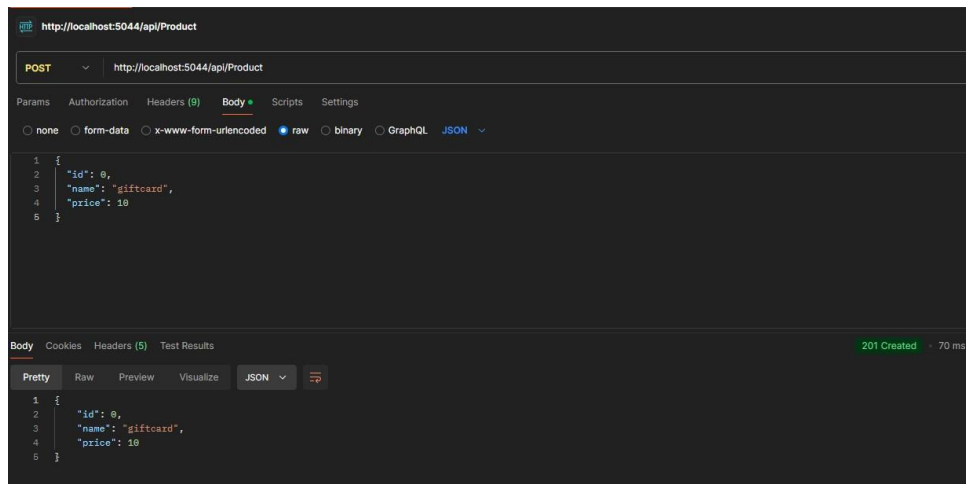
5. AB-05-User Login One parameter-Password: “password”



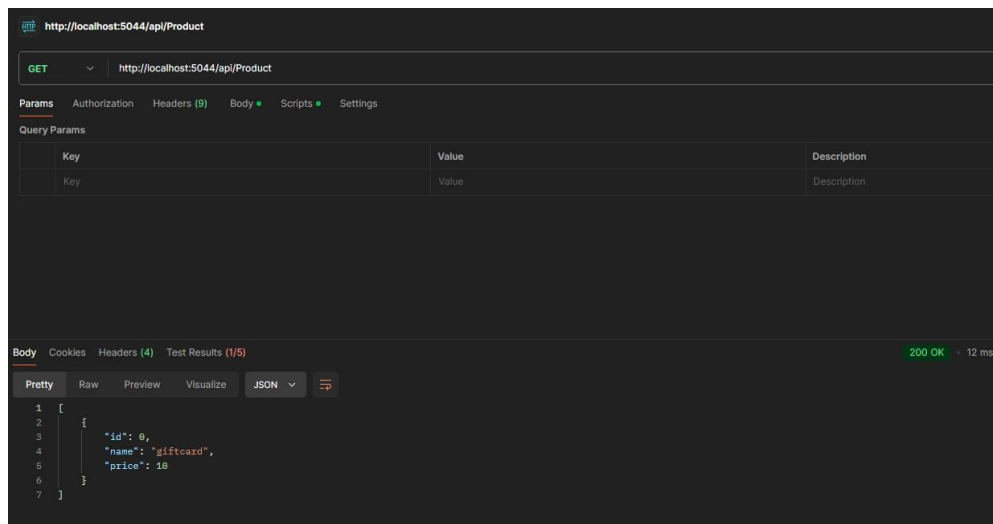
6. AB-06-User Login One parameter-UserName: “testuser”



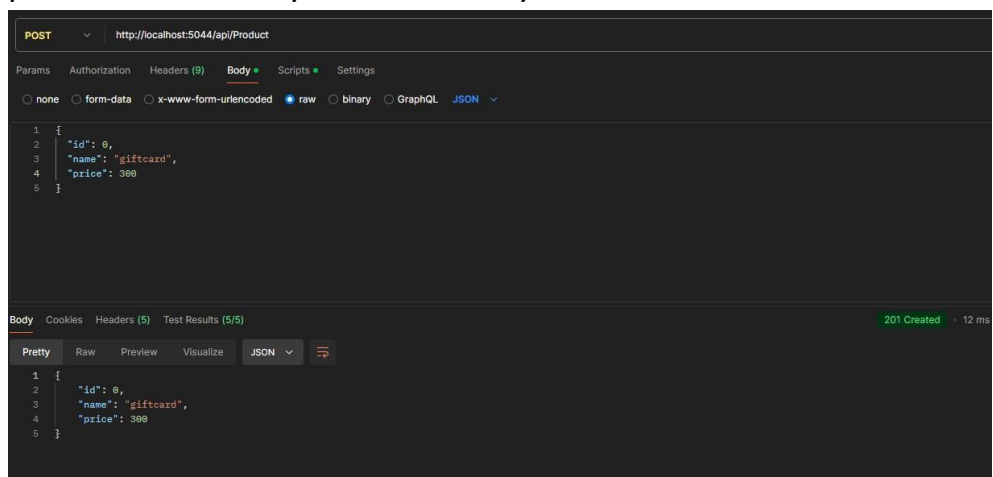
7. AB-07-the API allows to create a product with valid data



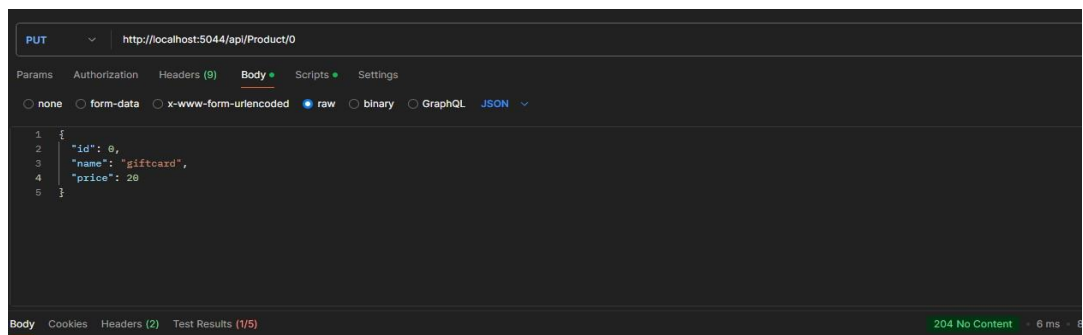
8. AB-08- the API allows consulting the list of products correctly



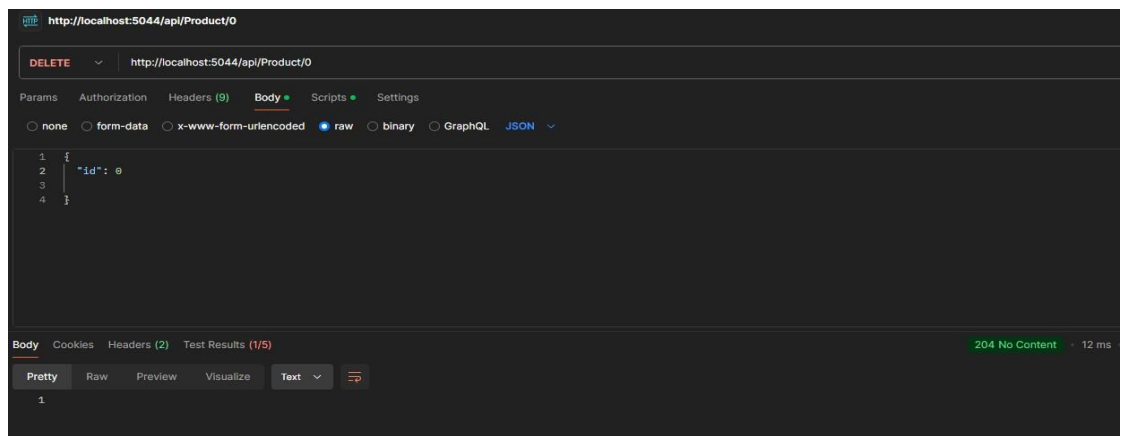
9. AB-09- the API does not allow the creation of a product if the provided ID already exists in the system



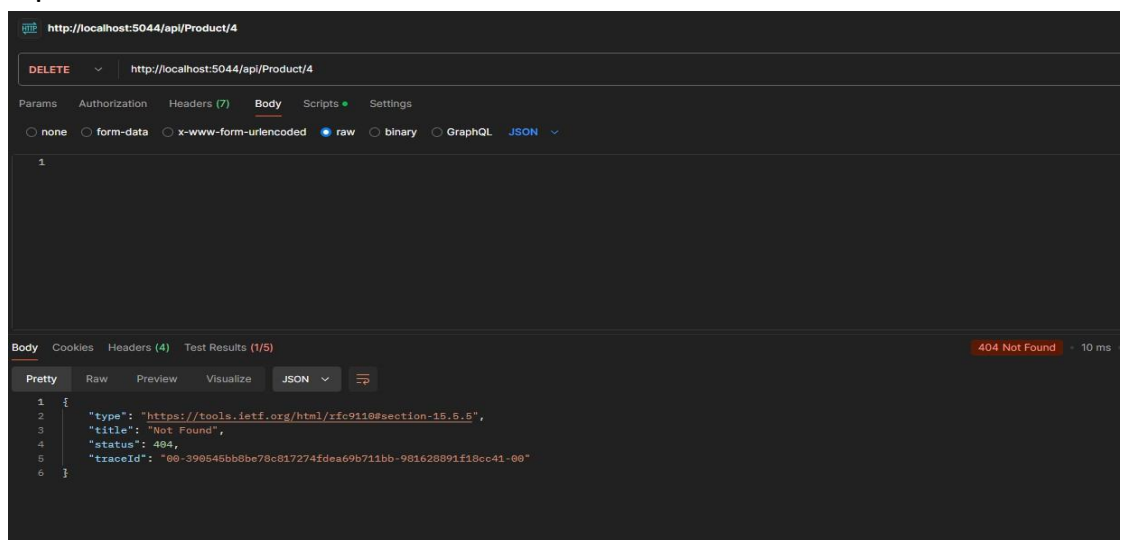
10.AB-10- the API allows editing a specific product using its ID.



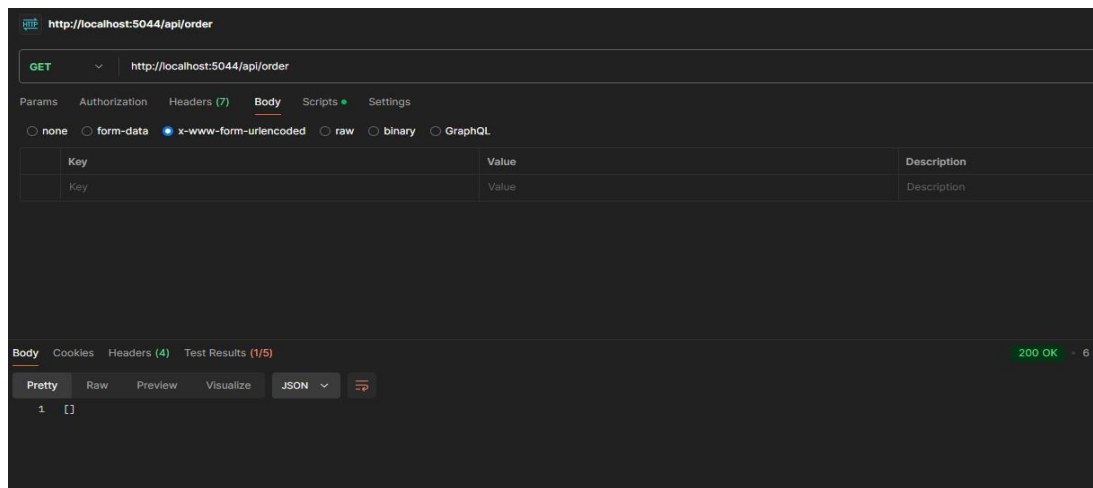
11.AB-11- the API allows you to delete a specific product using its ID.



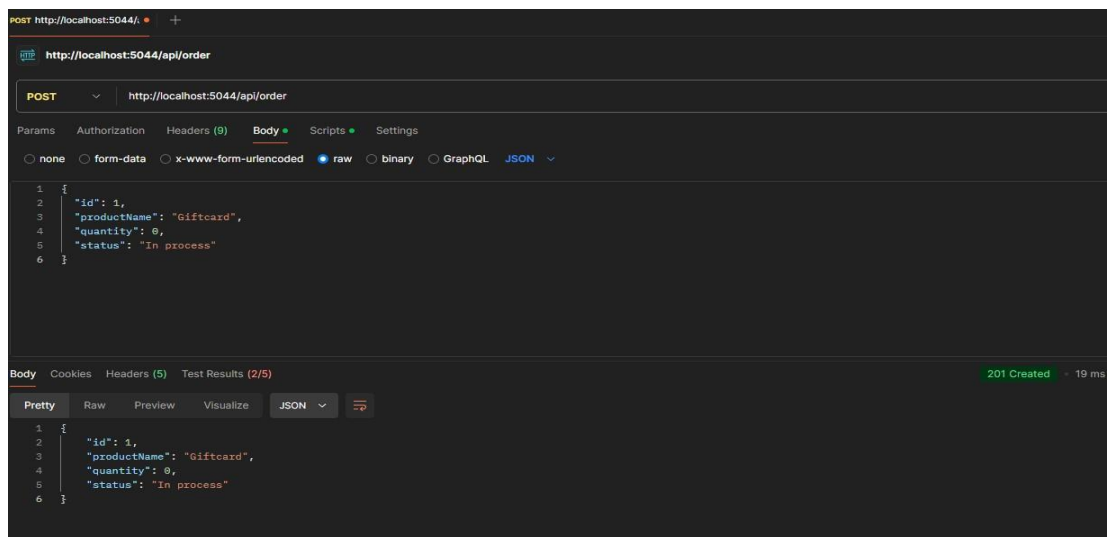
12.AB-12- the behavior of the system when an attempt is made to delete a product that does not exist.



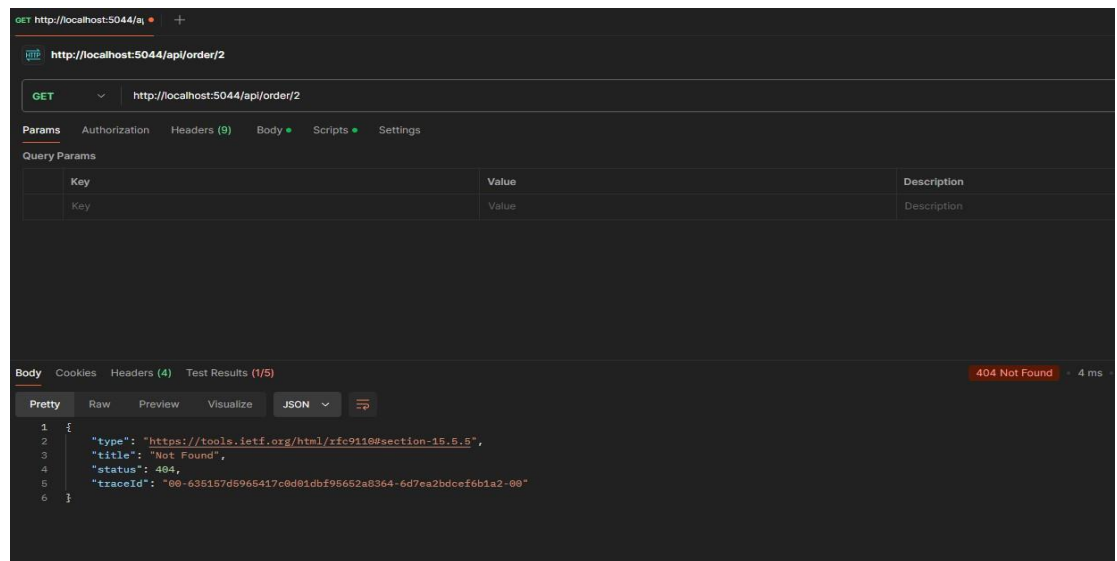
13.AB-13- the API correctly returns all existing orders.



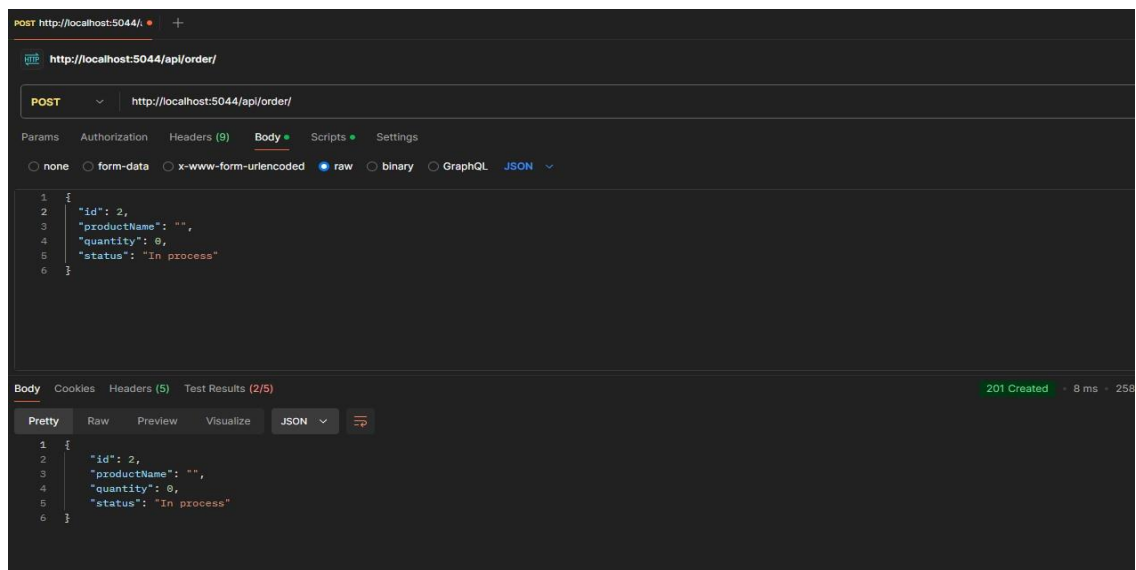
14.AB-14- the API correctly creates a specific order when provided with a valid ID



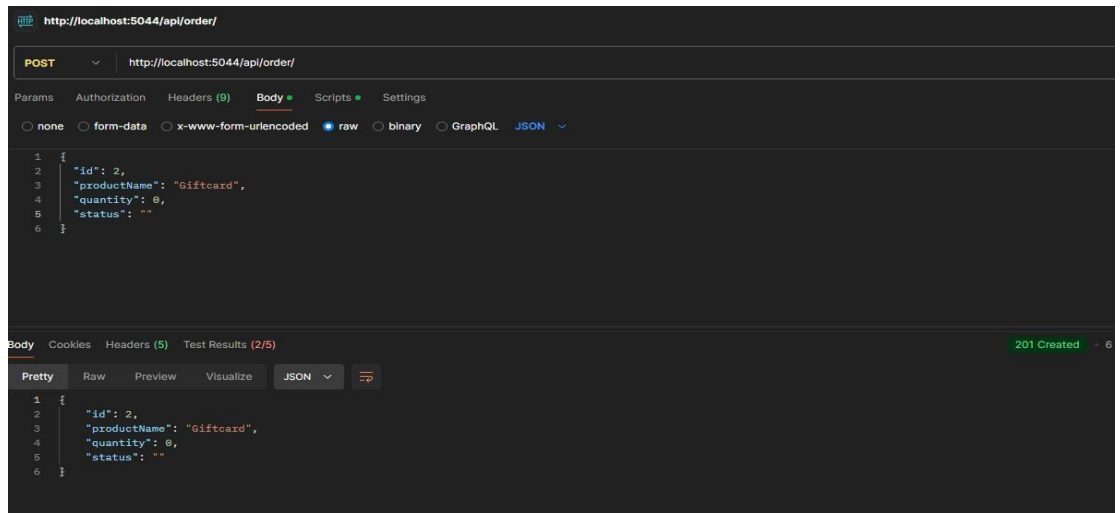
15.AB-15- the behavior of the API when requesting an order with an ID that does not exist.



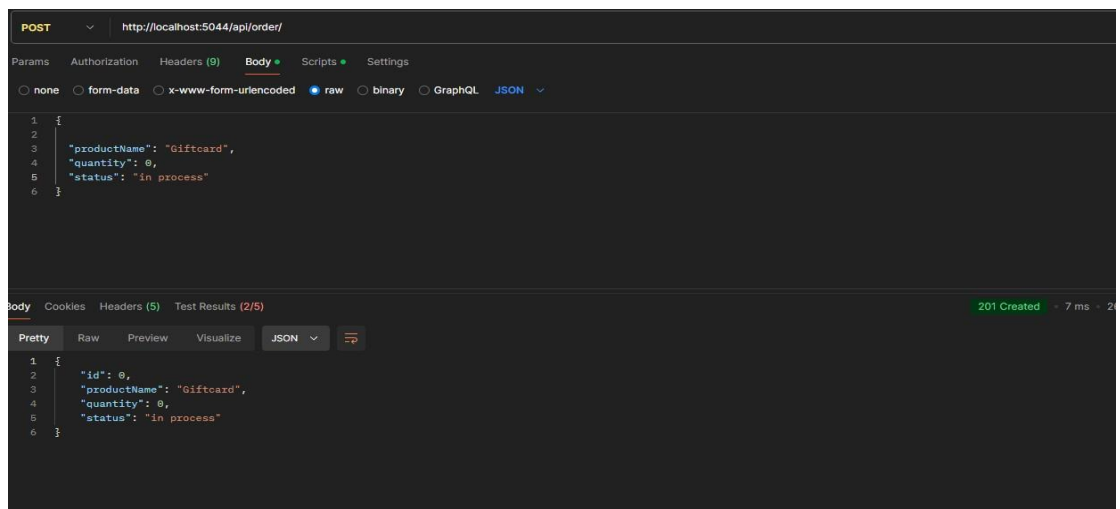
16.AB-16- the API properly handles the creation of an order when the productName is empty



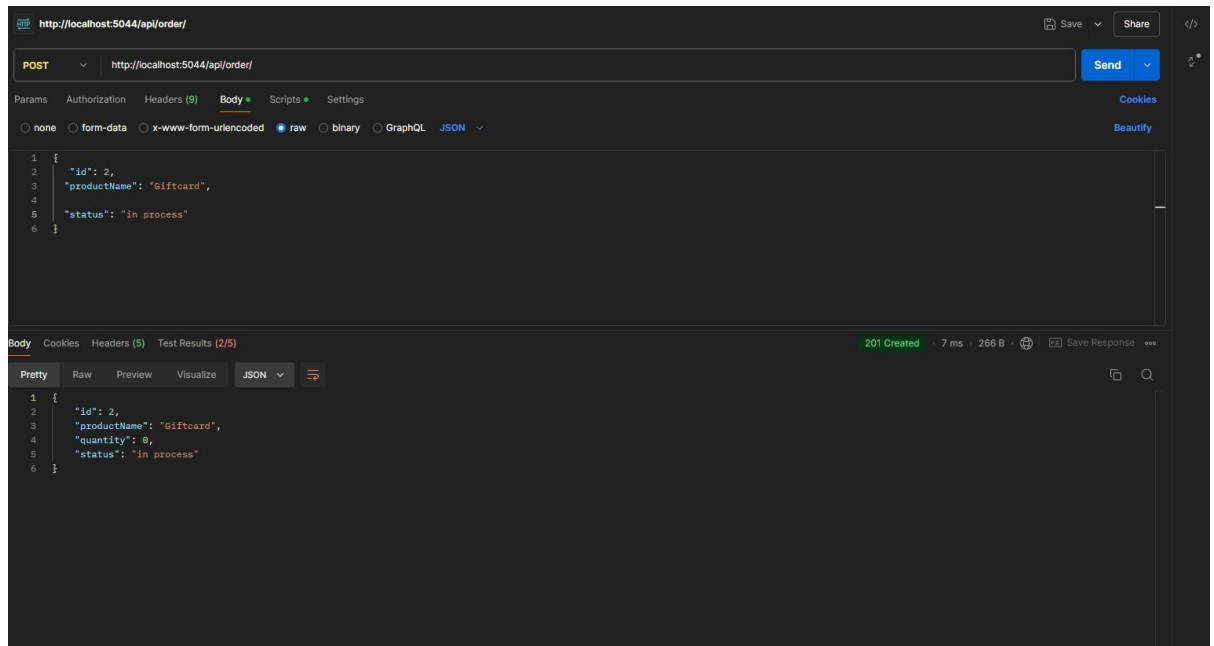
17.AB-17- the API properly handles the creation of an order when the status is empty



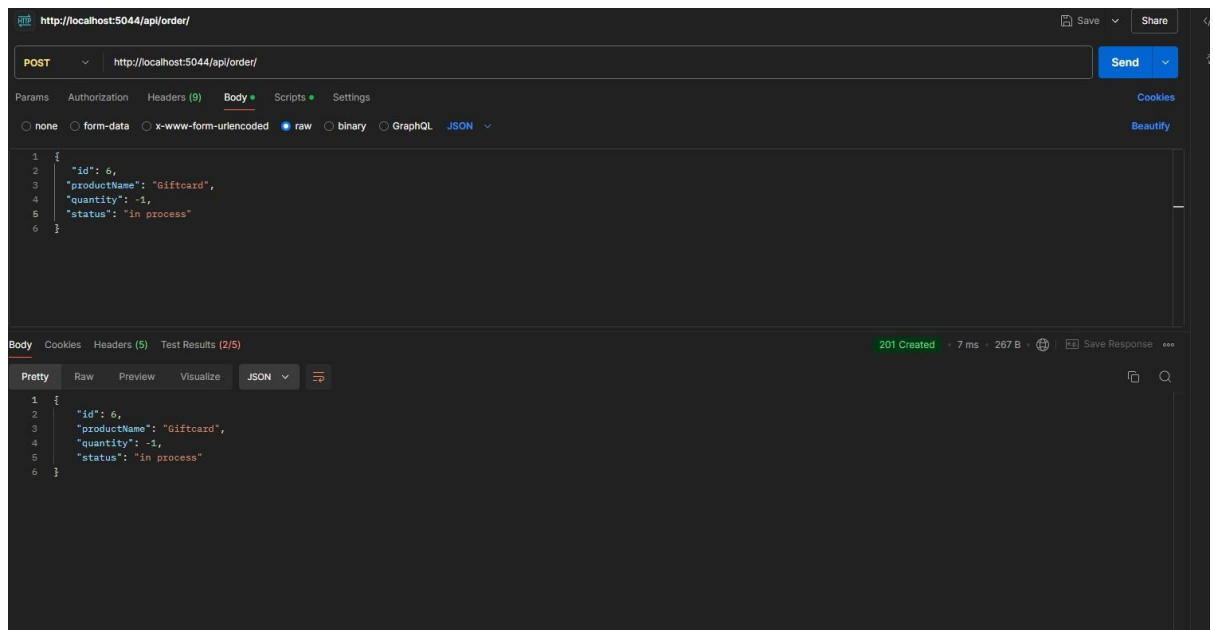
18.AB-18- the API properly handles the creation of an order when the ID is empty



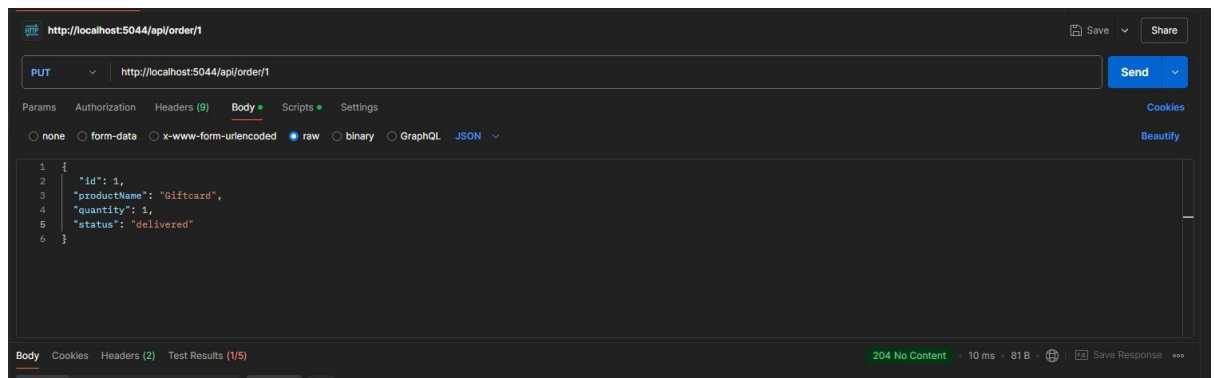
19.AB-19- the API properly handles the creation of an order when the Quantity is empty



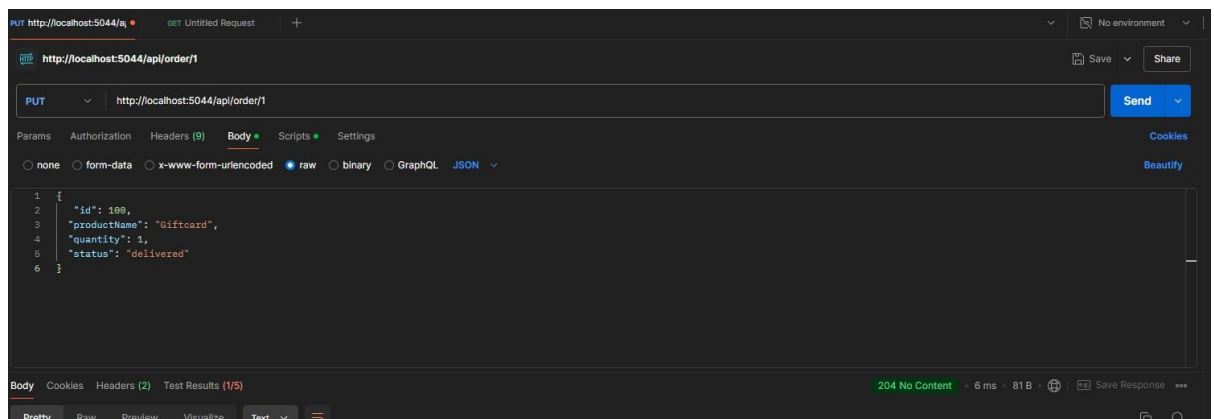
20.AB-20- the API rejects the creation of an order when quantity is negative.



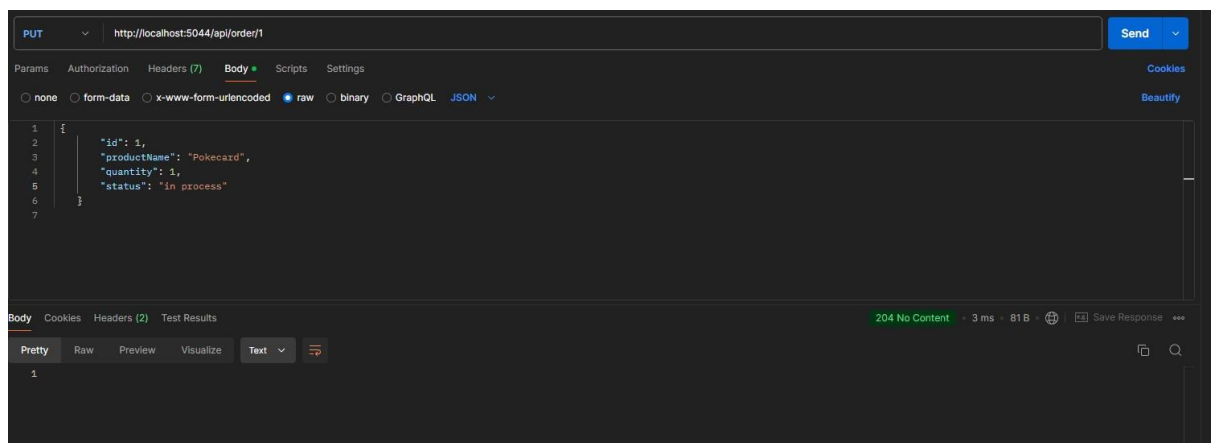
21.AB-21- Update order with valid data without changing the ID



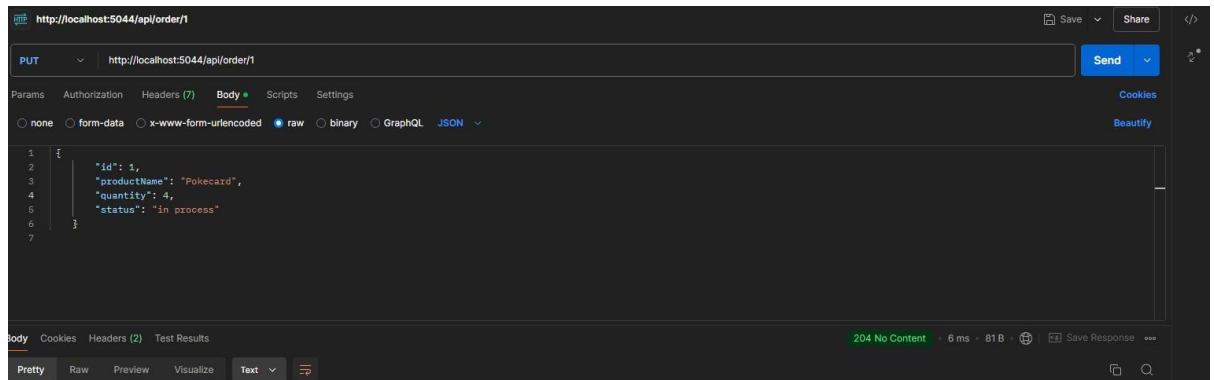
22.AB-22- Update the ID Of An Order



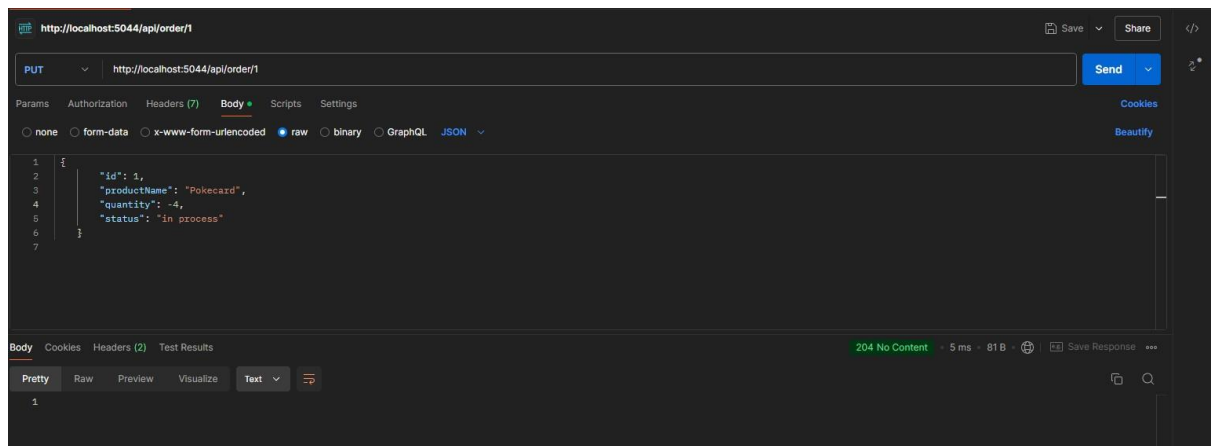
23.AB-23- the API properly handles the update of an order when the productName is other



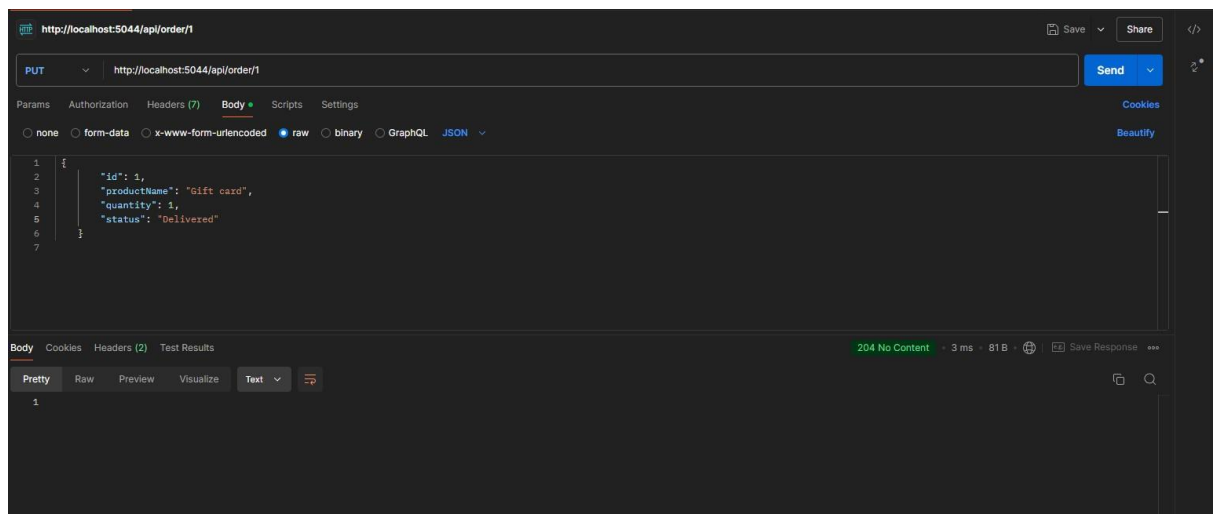
24.AB-24- the API properly handles the update of an order when the Quantity is other



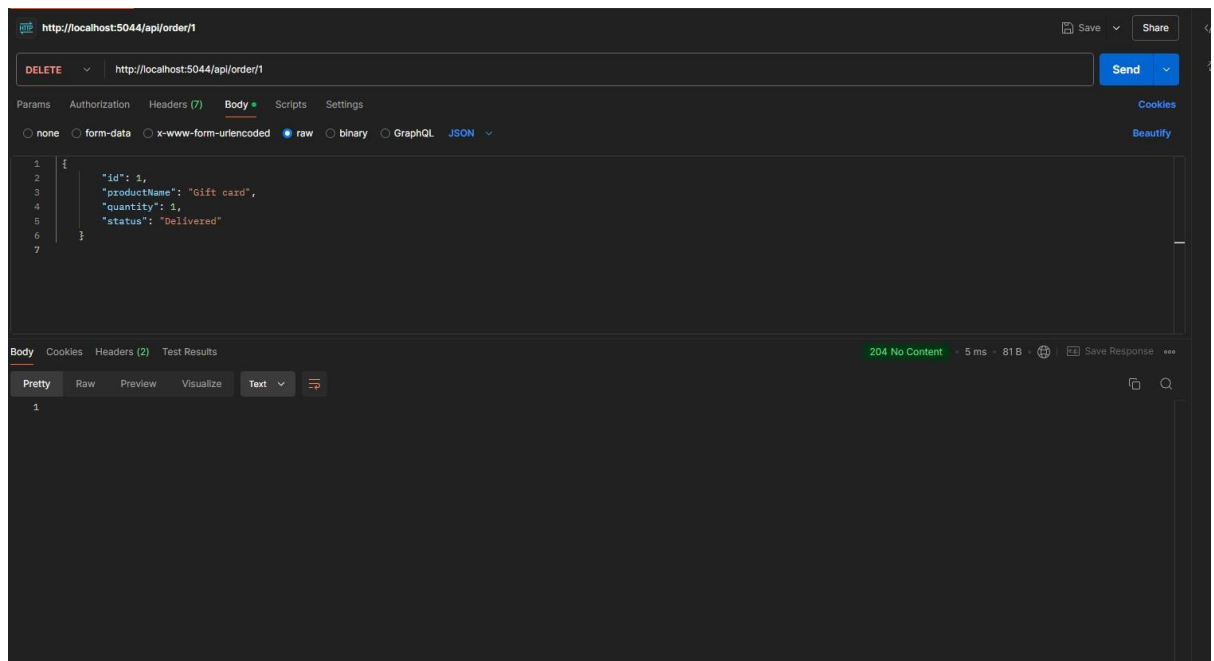
25.AB-25- the API properly handles the update of an order when the Quantity is negative



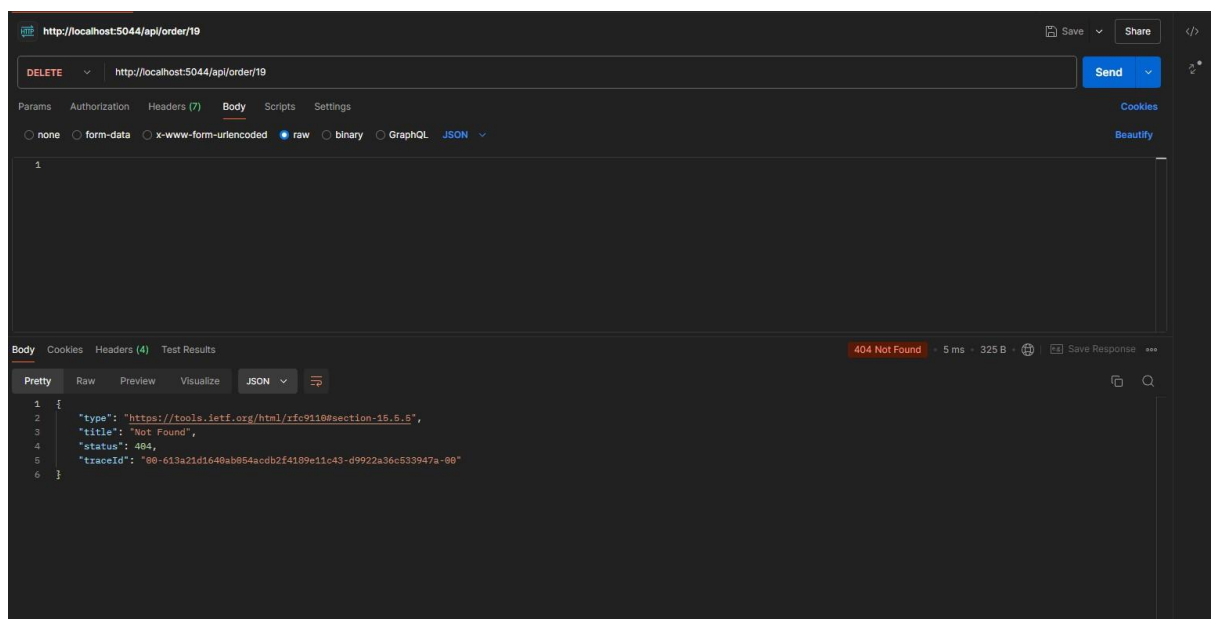
26.AB-26- the API properly handles the update of an order when the Status is other



27.AB-27- the API allows deleting an existing order with a valid order ID.



28.AB-28- Delete order with not existent ID



Documentation of Results:

Test Case ID	Test Case Title	Pass/Fail
AB-01	User login	Pass
AB-02	User login incorrect	Pass
AB-03	User Login Usurname Empty	Pass
AB-04	User Login password Empty	Pass
AB-05	User Login One parameter-Password: "password"	Pass
AB-06	User Login One parameter-UserName: "testuser"	Pass
AB-07	the API allows to create a product with valid data	Pass
AB-08	the API allows consulting the list of products correctly	Pass
AB-09	the API does not allow the creation of a product if the provided ID already exists in the system	Fail
AB-10	the API allows editing a specific product using its ID	Pass
AB-11	the API allows you to delete a specific product using its ID.	Pass
AB-12	the behavior of the system when an attempt is made to delete a product that does not exist	Pass
AB-13	the API correctly returns all existing orders	Pass
AB-14	the API correctly creates a specific order when provided with a valid ID	Pass
AB-15	the behavior of the API when requesting an order with an ID that does not exist	Pass
AB-16	the API properly handles the creation of an order when the productName is empty	Fail
AB-17	the API properly handles the creation of an order when the status is empty	Fail

AB-18	the API properly handles the creation of an order when the ID is empty	Fail
AB-19	the API properly handles the creation of an order when the Quantity is empty	Fail
AB-20	the API rejects the creation of an order when quantity is negative	Fail
AB-21	Update order with valid data without changing the ID	Fail
AB-22	Update the ID Of An Order	Fail
AB-23	the API properly handles the update of an order when the productName is other	Fail
AB-24	the API properly handles the update of an order when the Quantity is other	Fail
AB-25	the API properly handles the update of an order when the Quantity is negative	Fail
AB-26	the API properly handles the update of an order when the Status is other	Pass
AB-27	the API allows deleting an existing order with a valid order ID	Pass
AB-28	Delete order with not existent ID	Pass

Defect Identification:

ID	Def-01
Severity	High
Title	There is no way to log out of the website.
Test case	AD-01
Description	After successfully logging in, the user cannot log out of the website, since there is no button or link to access it
Steps	In userName: "testuser" In Password: "password", Click the Login button.
Result	Nothing happens; only a 'sample Token' message is displayed
Expected Result	You would successfully log in to your account, and a visible logout button would be available

ID	Def-02
Severity	Medium
Title	The system not display a specific error message
Test case	AB-03 and AB-04
Description	The system does not display a specific error message for empty fields in the login.
Steps	1. Access the login page. 2. Leave both fields empty, only fill in the user name, or only fill in the password. 3. Click on "Login"
Result	The system returns a 401 (Unauthorized) error.
Expected Result	The system should display a specific message indicating that all fields are mandatory.

ID	Def-03
Severity	High
Title	Access to orders without logging in.
Test case	PO-07
Description	The user can access the order list without being logged in
Steps	Click on the Order link.
Result	It should not allow access to the orders without being logged in.
Expected Result	The user is allowed to access the orders.

ID	Def-04
Severity	High
Title	Allows setting the same ID twice for the same product
Test case	AB-09
Description	The system allows the creation of a product with an ID that already exists, which should not be allowed.
Steps	1. Send a POST request to /api/Product with an existing product ID. 2. Observe the error system
Result	The system returns a 201 status
Expected Result	The system should return an error with a 409 status and a message indicating that the ID already exists.

ID	Def-05
Severity	Medium
Title	API allows the creation of an order with negative quantity
Test case	AB-20
Description	Allows setting a negative quantity
Steps	1. Send a POST request to /api/Order with the amount in negative and all other valid data.
Result	A 201 status was received and the order was created with negative quantity.
Expected Result	The API should reject the order creation and return a status 400 or 422, indicating that quantity cannot be negative

ID	Def-06
Severity	High
Title	Order created without name field.
Test case	AB-16
Description	When creating a Order without entering the name field, it is created successfully.
Steps	Add the required characters and leave the name field empty
Result	The Order is created successfully.
Expected Result	A Order without a name should not be able to be created.

ID	Def-07
Severity	High
Title	Order created without the Status field
Test case	AB-17
Description	When trying to create a new order without the Status field, it is created successfully.
Steps	Add the required characters and leave the Status field empty
Result	The order is created even without Status field
Expected Result	The response has the status code 400 Bad Request, the body shows error information and the new product cannot be created

