

Retrieving the full dataset works well and in a successful request returns a status code 200 as it should.

However, we believe error handling could have been implemented better, so the server could give status code, such as 400, in an event of a wrong user request (e. g. requesting with non-empty body).

The screenshot displays a web browser's developer tools interface. The top section shows a GET request to `http://localhost:3000/photos` with a status of 200 OK, a time of 93 ms, and a size of 963 B. The response body is a JSON array of two photo objects. The first object is for Tim Berners-Lee, and the second is for Grace Hopper. The bottom section shows the console with two GET requests, both returning 200 status codes.

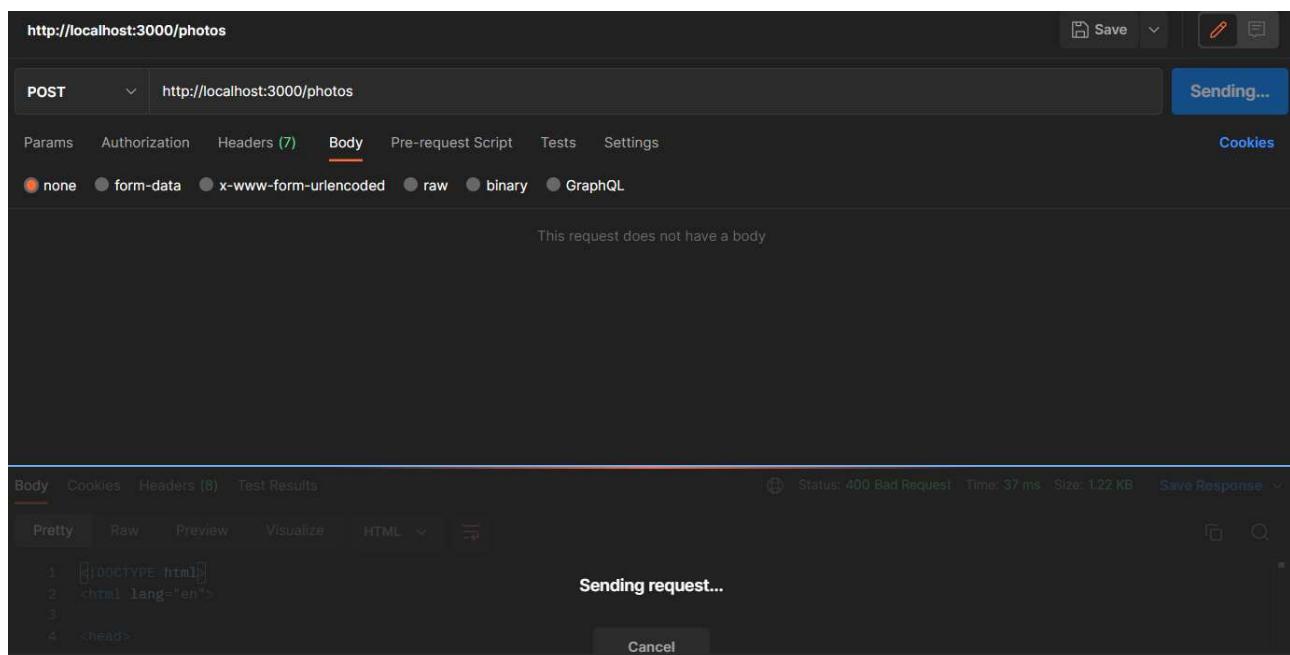
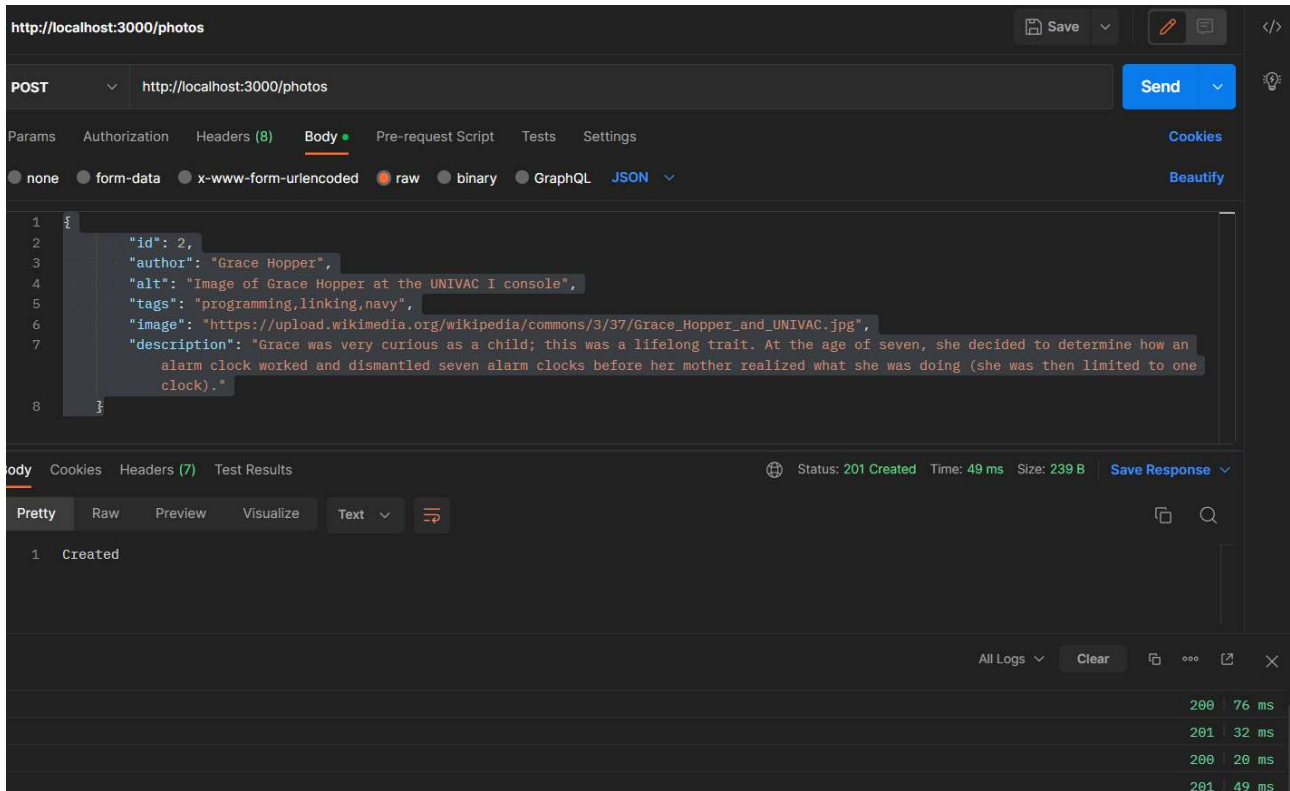
```
1 {
2   {
3     "id": 1,
4     "author": "Tim Berners-Lee",
5     "alt": "Image of Berners-Lee",
6     "tags": "html,http,url,cern,mit",
7     "image": "https://upload.wikimedia.org/wikipedia/commons/9/9d/Sir_Tim_Berners-Lee.jpg",
8     "description": "The internet and the Web aren't the same thing."
9   },
10  {
11    "id": 2,
12    "author": "Grace Hopper",
13    "alt": "Image of Grace Hopper at the UNIVAC I console",
14    "tags": "programming,linking,navy",
15    "image": "https://upload.wikimedia.org/wikipedia/commons/3/37/Grace_Hopper_and_UNIVAC.jpg",
16    "description": "Grace was very curious as a child; this was a lifelong trait. At the age of seven, she decided to determine how an
17                  alarm clock worked and dismantled seven alarm clocks before her mother realized what she was doing (she was then limited to one
18                  clock)."
```

Console logs:

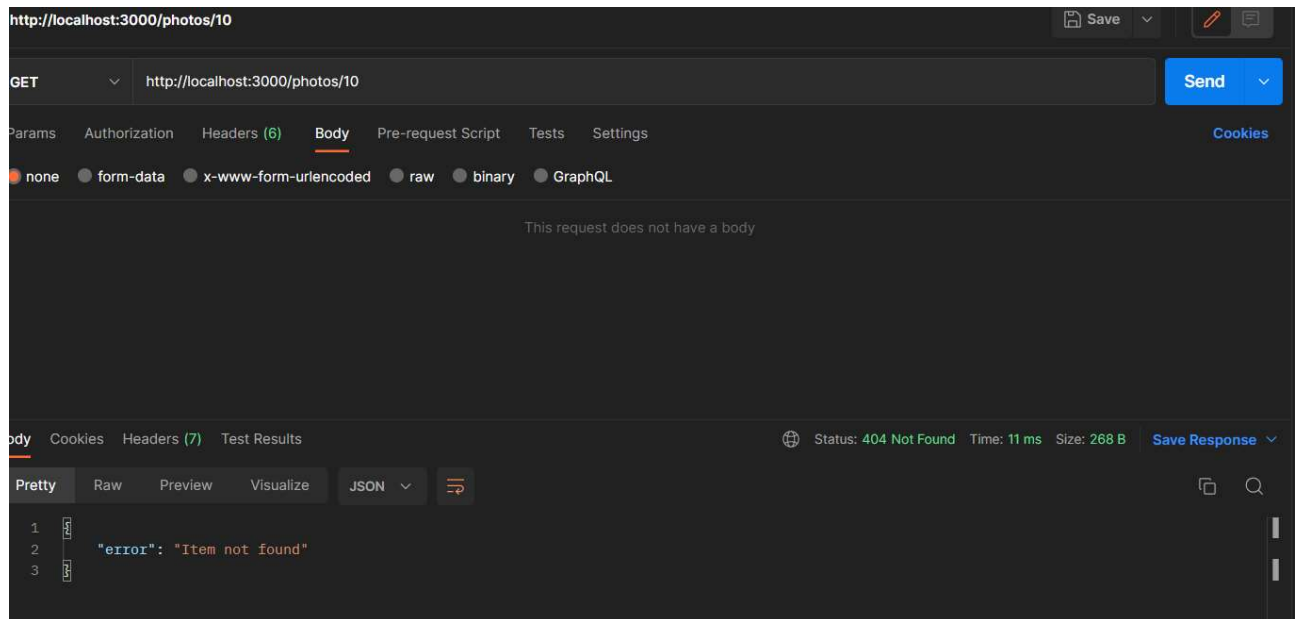
Method	URL	Status	Time
GET	http://localhost:3000/photos	200	93 ms
GET	http://localhost:3000/photos	200	76 ms

Adding data with proper request body works perfectly well and gives the correct response code 201. However, when the request body is empty or at least one of the mandatory inputs in request body (such as “author” or “image”, etc.) is missing, the server fails to give error code and gets stuck on loading screen.

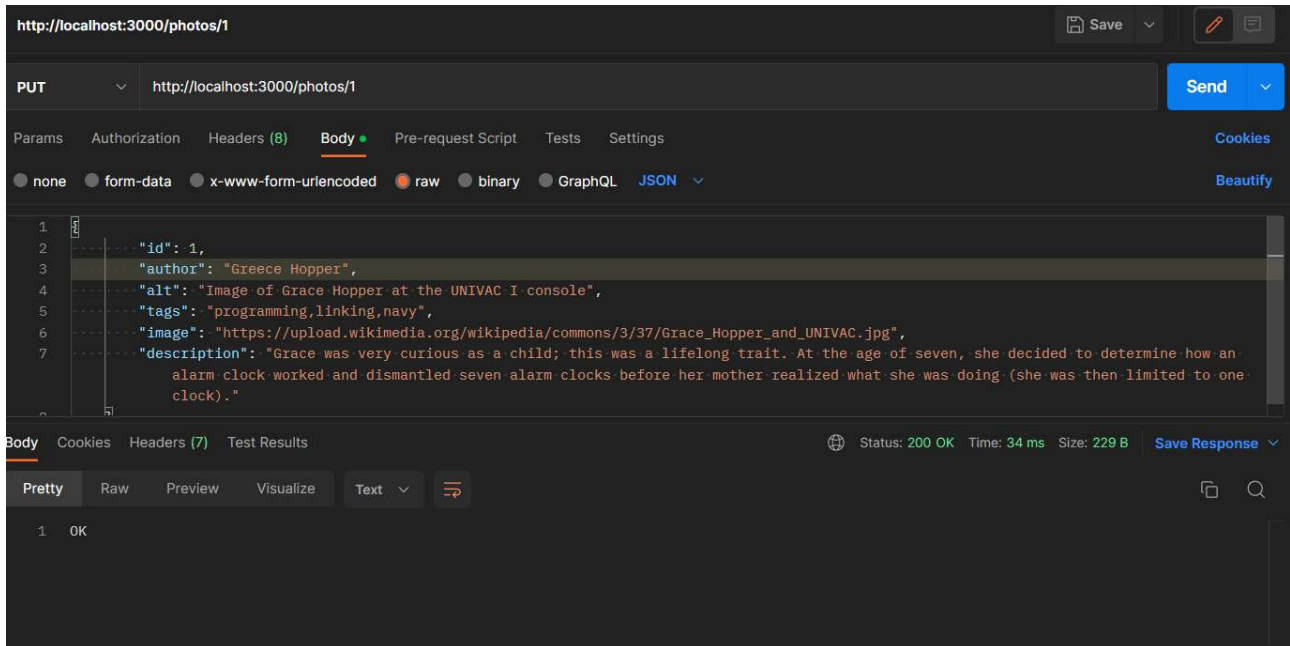
When the request body element is not missing, but the user leaves its value blank (“author”: “”), the server responds correctly with status code 400.



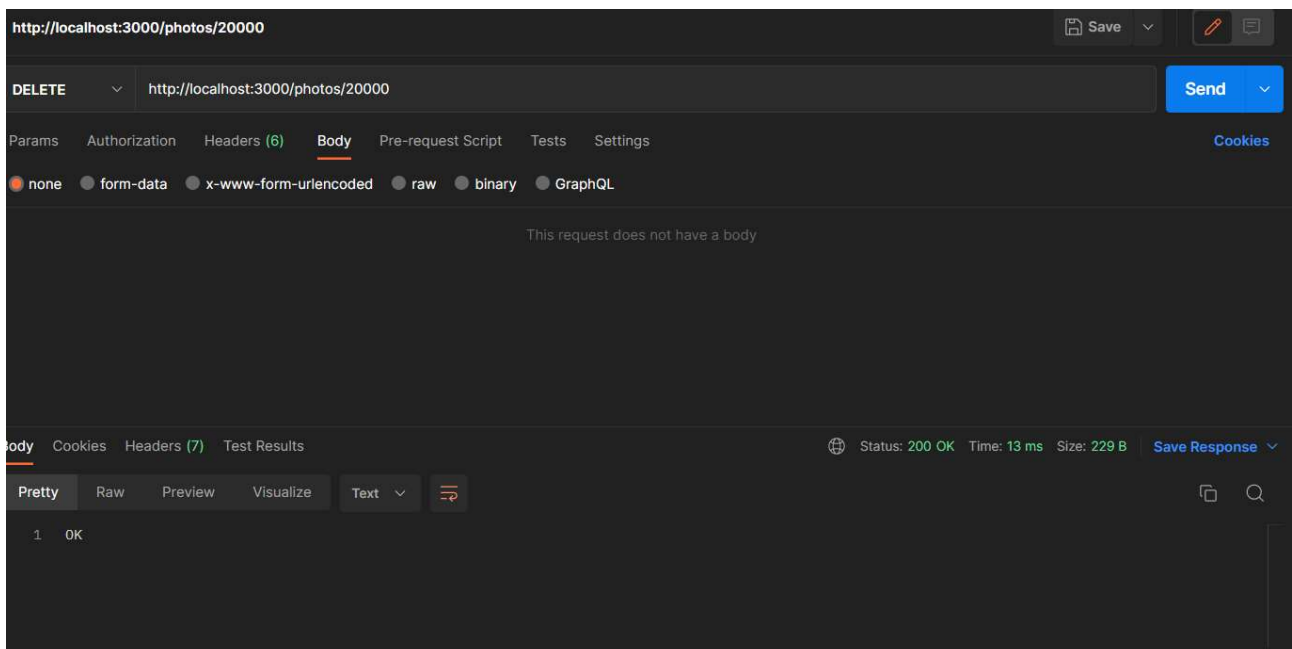
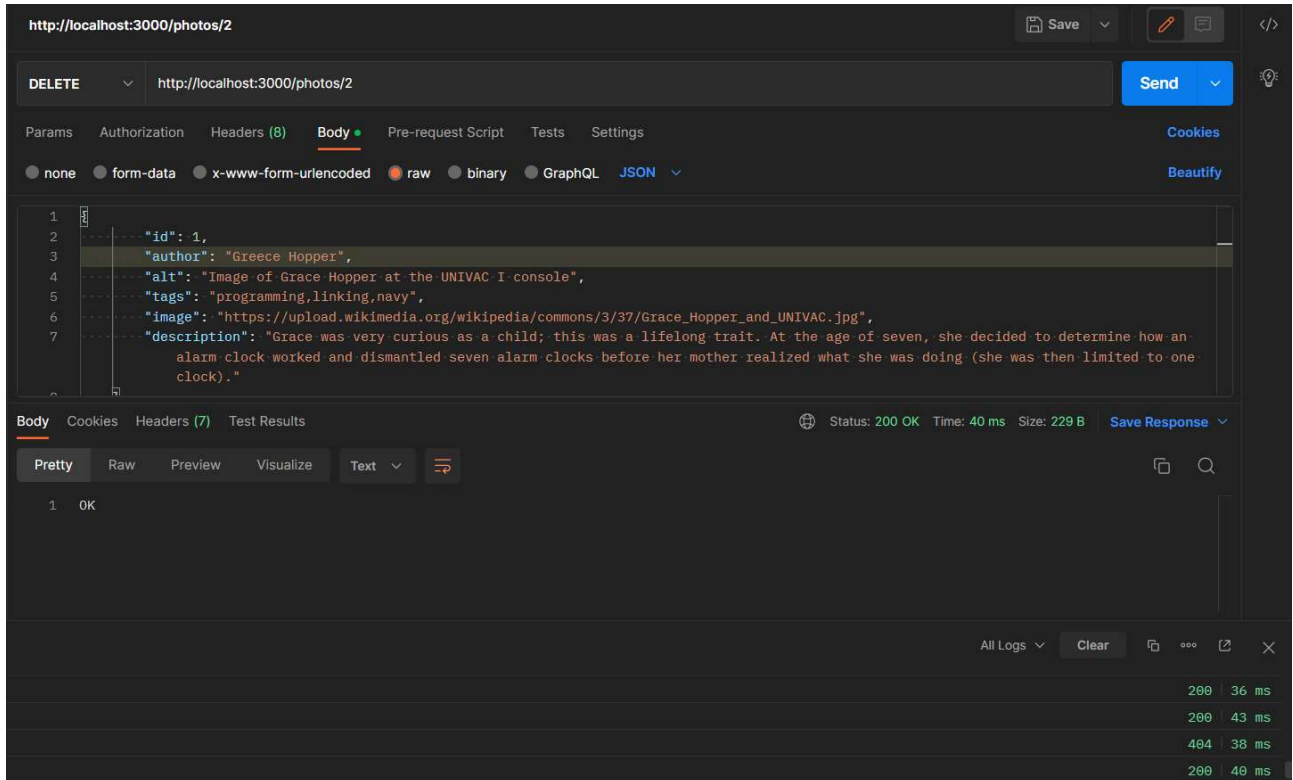
Listing data for specific item works very nicely and we can only congratulate the group for achieving it. When the id is found in the database, the server returns proper data with status code 200, and when the id is not correct, the response is an error message with status 404.



Changing data for a specific item works well too. When the information is updated, we get a status code 200 and after checking if the change has actually happened by using GET method, we see that it did. We suggest adding more error handling here though for situations, such as user providing improper data to update with error 400 or user trying to update image with non-existent id with error code 404.



Finally, removing a specific item works fine when the item is present, but we still have a couple suggestions for improvement. Firstly, the success code could be 204 rather than 200 just so you use more codes in appropriate places. Secondly, trying to remove an item that has already been removed/doesn't exist still gives status code 200 (image 2), while it should be code 404 or code 400.



The bonus has not been implemented.

Overall, the group clearly understands how to construct a server with all the required functionality working well. For the positives, the group's documentation is very understandable and makes it easy to use the API, the server is nicely responsive and does what it is supposed to and the code itself is easy-to-read. However, there is still space for improvement in using proper response codes and paying attention to every possible users' input to adjust the way server reacts to it.