Homework 1

Elena Bucchianeri, 463889

20 ottobre 2017

The service is located at: https://murmuring-peak-81684.herokuapp.com/
The messages shown below are listed according to URI and method. Some of these messages are saved in the postman collection file and they can be tested in the order specified in the collection to have the same output.

1 /doodles (GET)

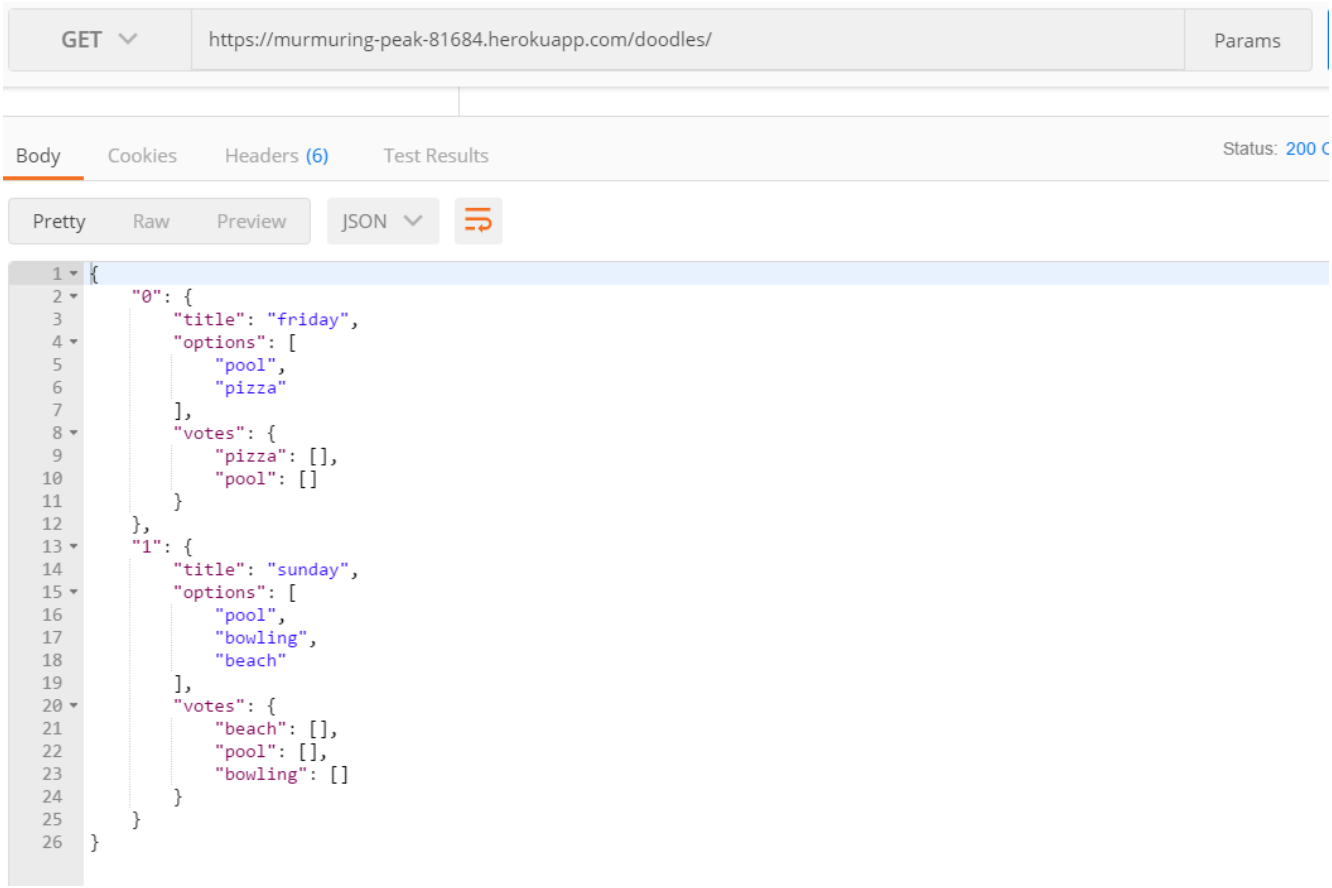

Figura 1: GET request in case of non empty doodle hashmap. The service returns doodles turned into a JSON Object.

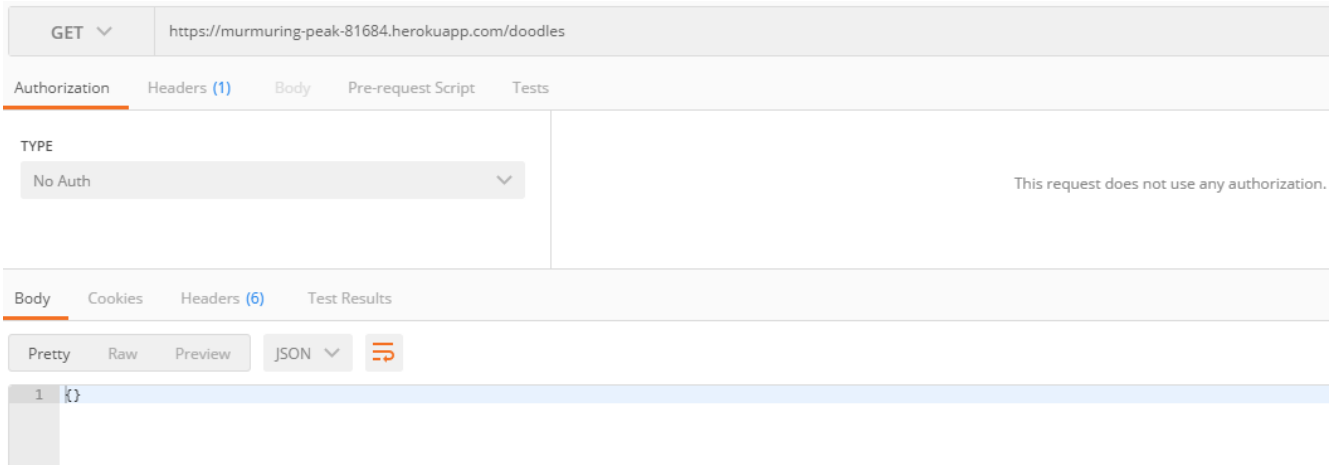

Figura 2: GET request in case of empty doodle hashmap. Service returns an empty object.

2 /doodles (PUT)

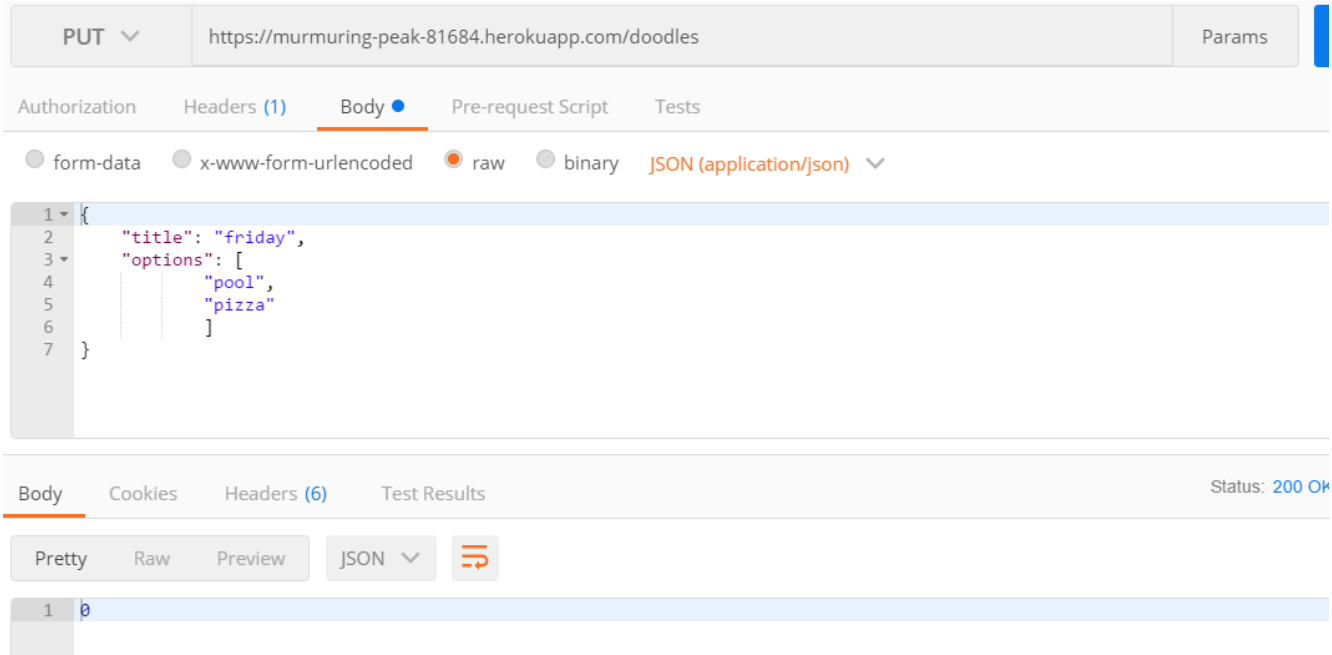

Figura 3: Insert request (PUT) of a new doodle. Service returns ID associated to the doodle. After the first PUT, if this PUT request is repeated the outcome is always the same.

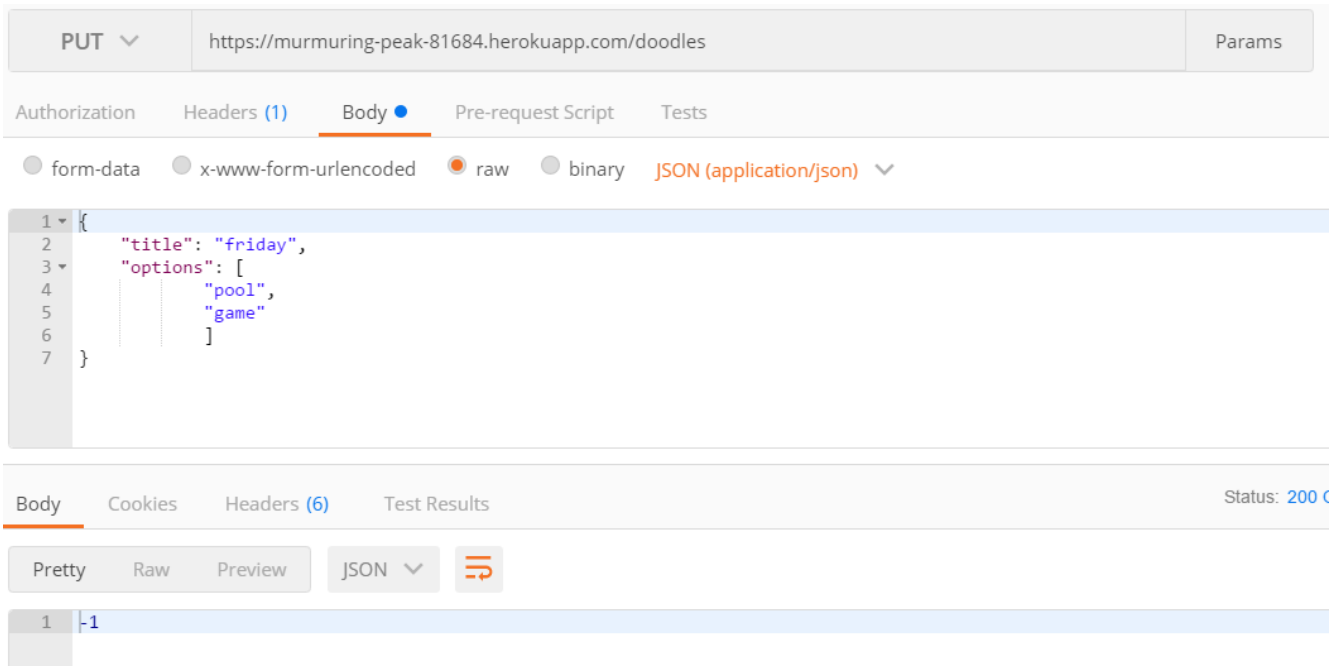

Figura 4: Insert request (PUT) of a doodle. There is already in the hashmap a doodle called "friday" (as in Figure 1 with different options, so the new doodle cannot be created. The service returns an error code.

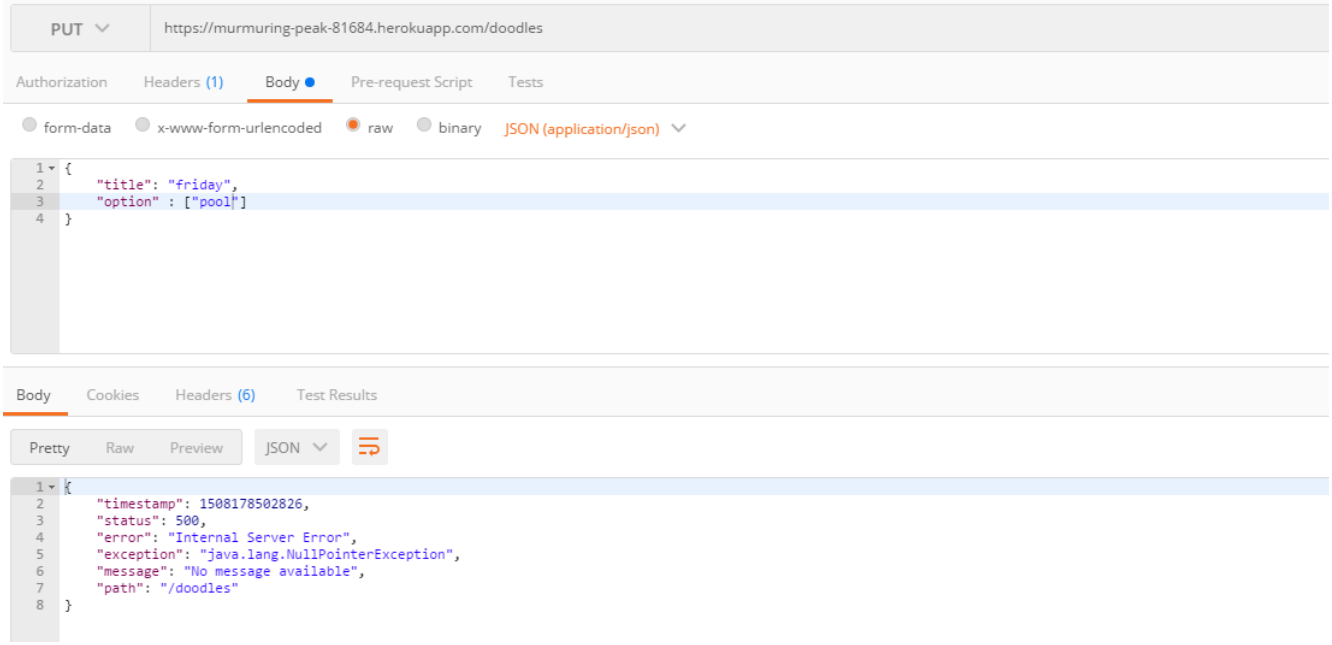

Figura 5: PUT request of a wrong JSON Doodle. Service returns an error

3 /doodles/{id}(GET)

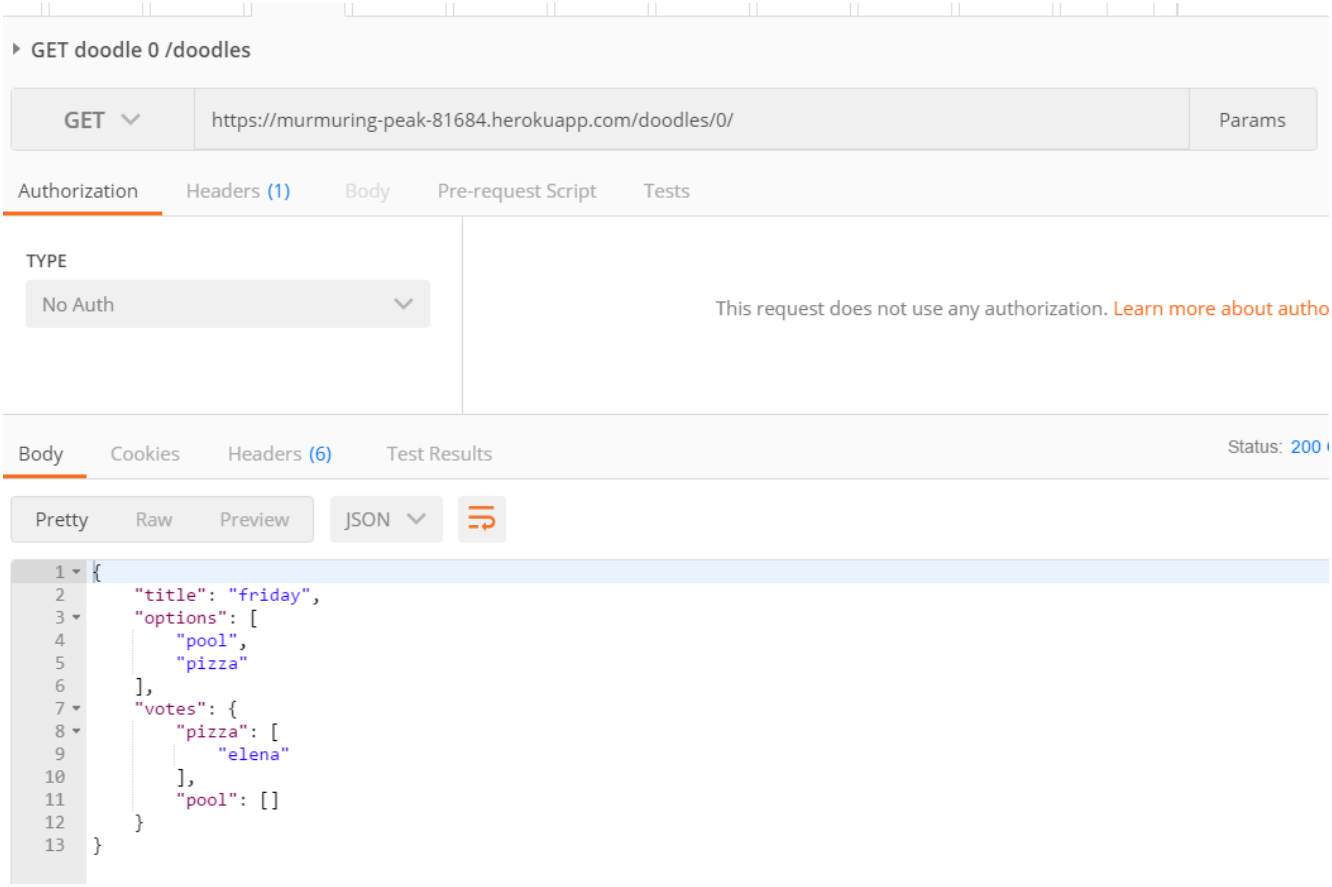

Figura 6: Correct GET request of a Doodle by ID. Service returns the Doodles associated to this ID.

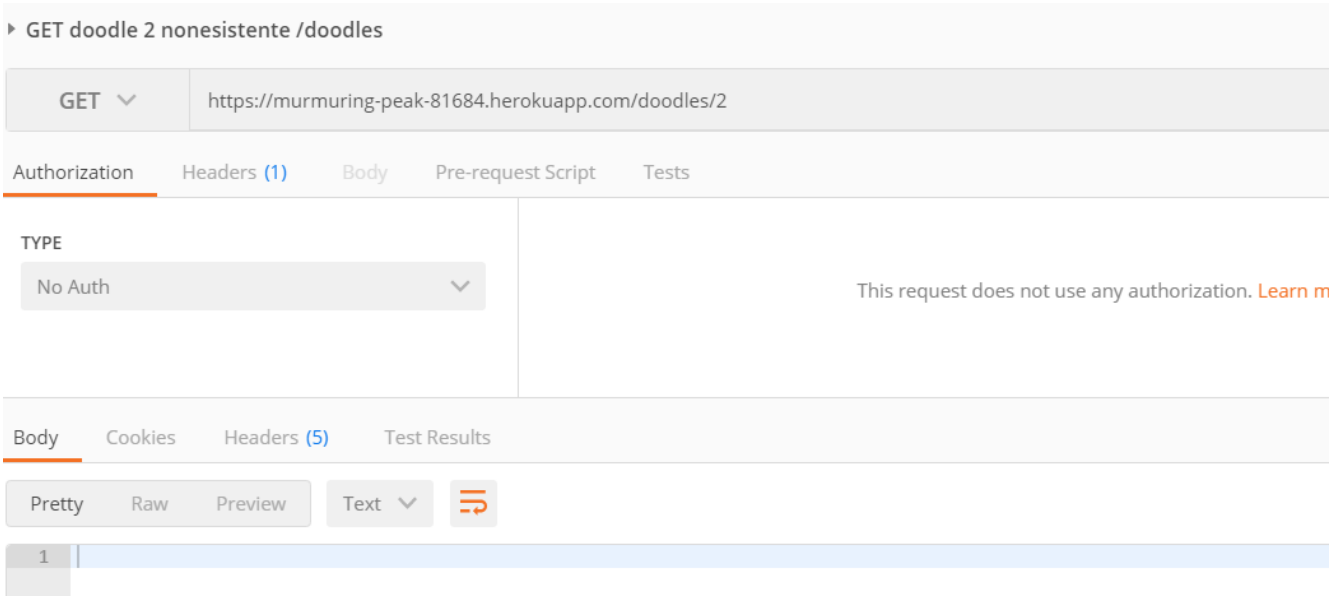

Figura 7: GET request of a Doodle by ID when ID does not exist. In this case the service replies with null.

4 /doodles/{id}(DELETE)

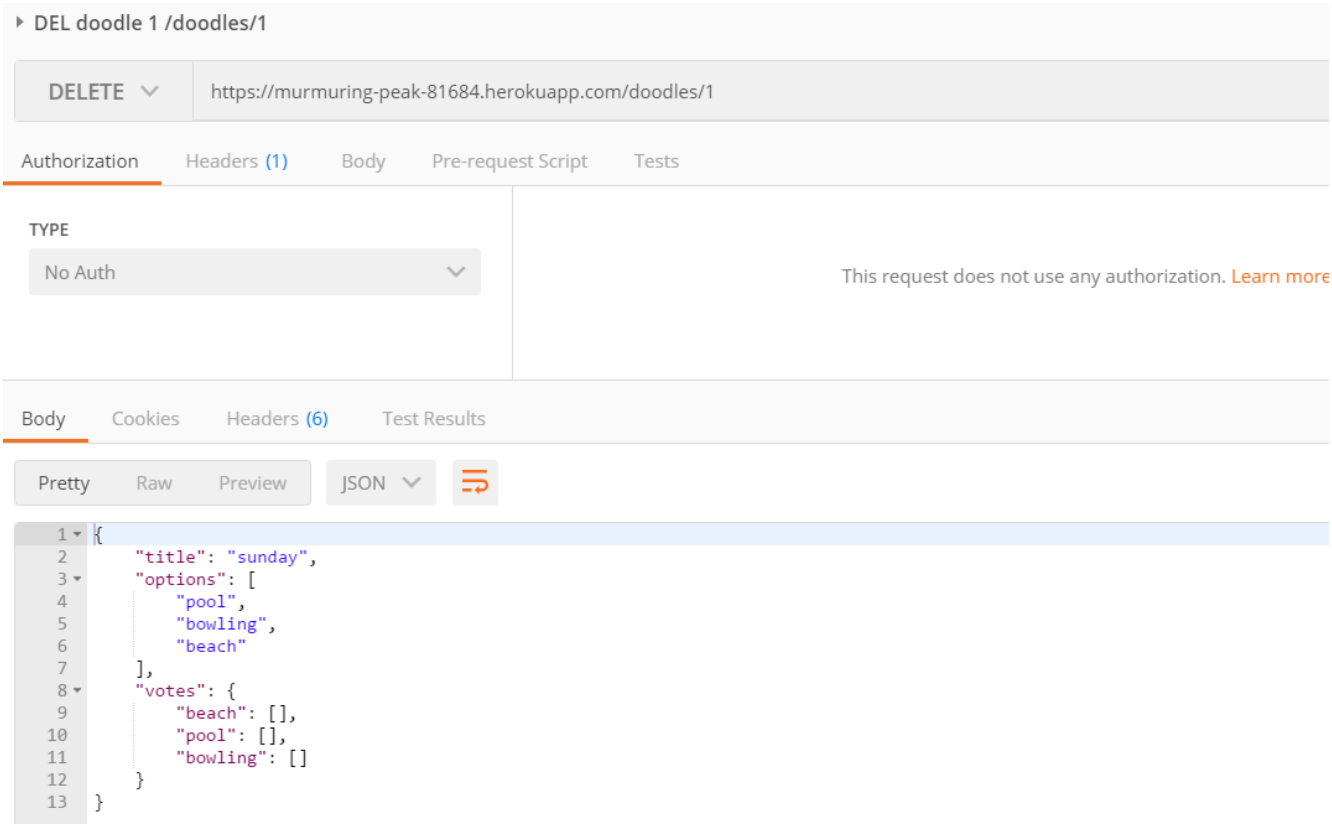

Figura 8: Correct DELETE request of a Doodle by ID. Service replies with the Doodle associated to this ID.

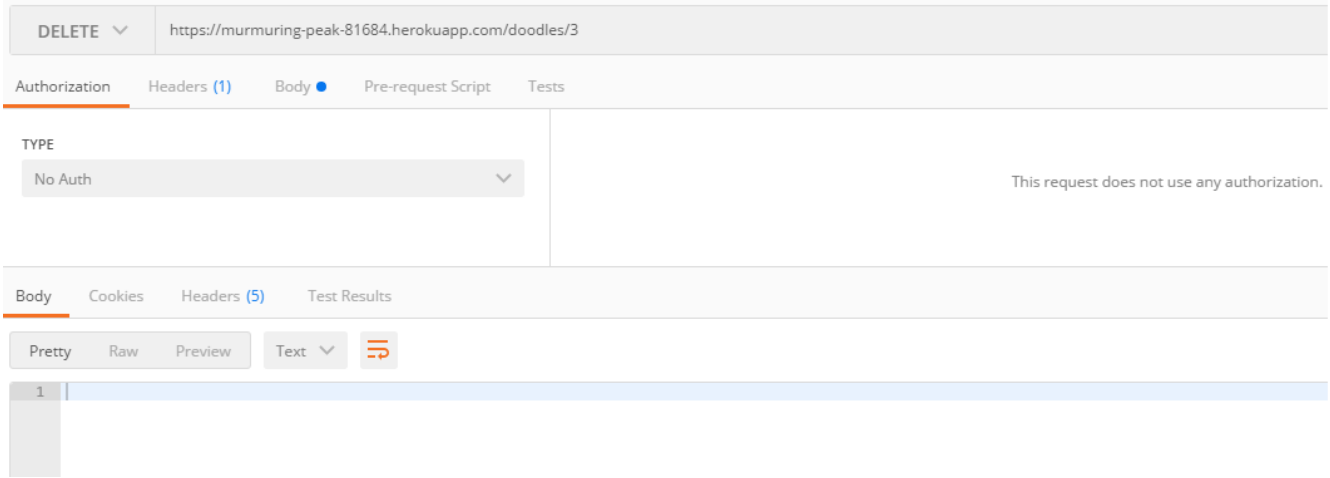

Figura 9: DELETE request of a Doodle by ID when ID does not exist. In this case the service returns null.

5 /doodles/{id}/vote (PUT)

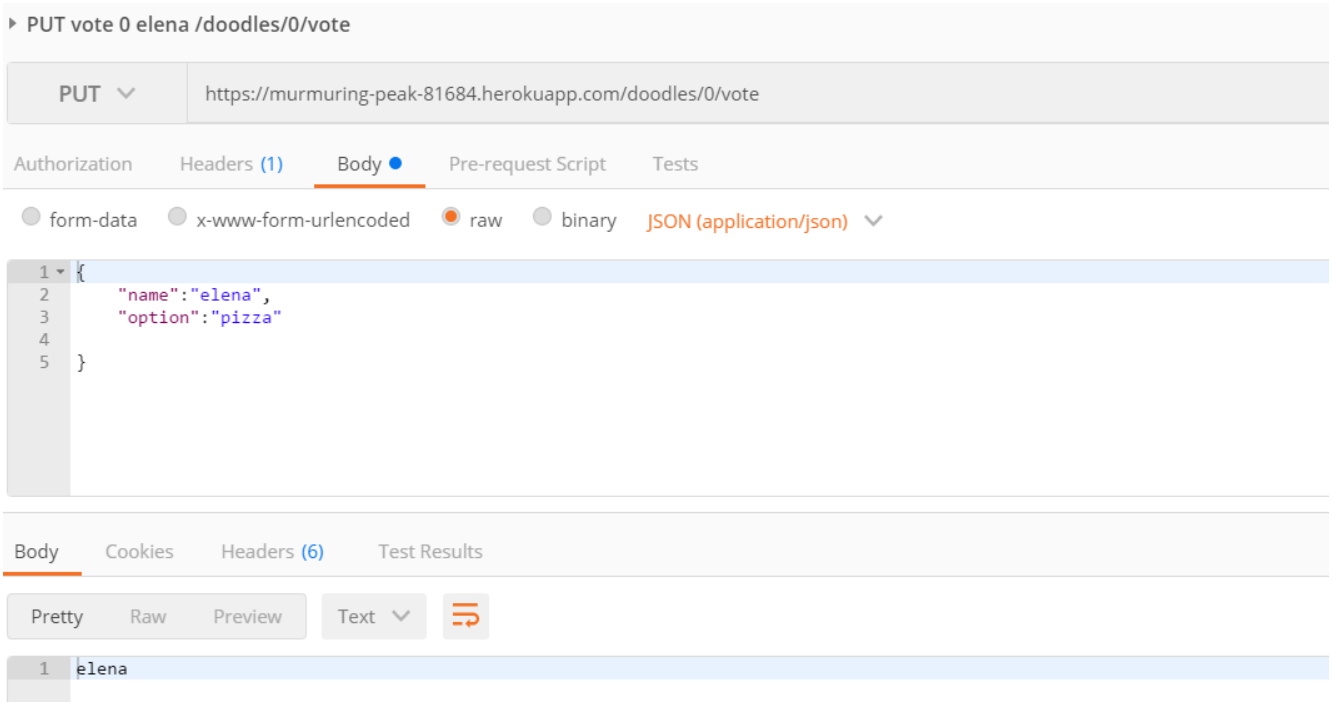

Figura 10: Correct PUT request. Service replies with the name of the voter.

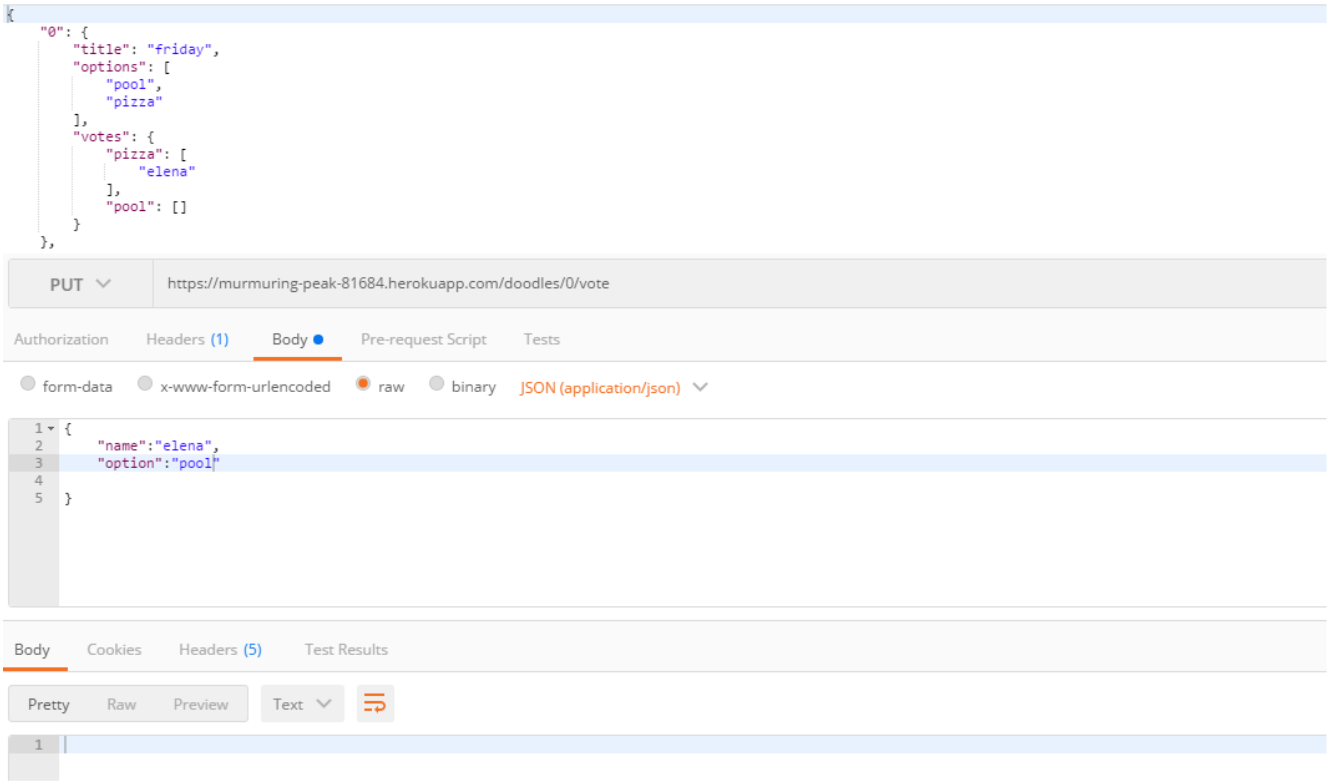

Figura 11: PUT request of a vote. This user has already voted an other option of this doodle, so this new vote is not considered and service replies with null.

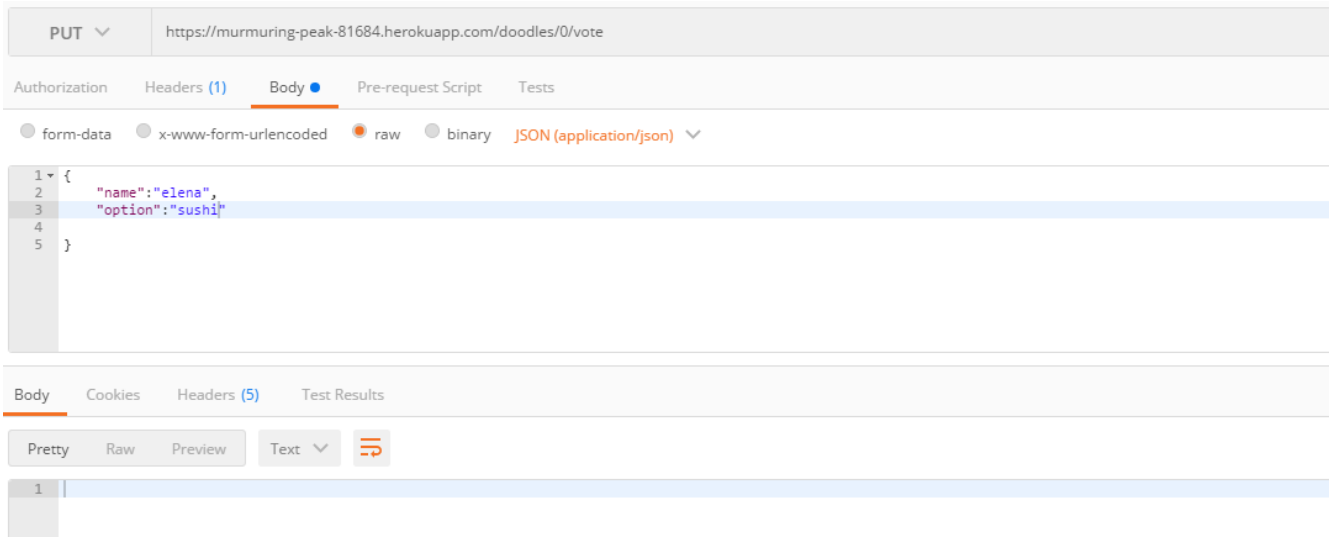

Figura 12: PUT request of a vote with a wrong option (doodle 0 has not "sushi" option, as in Figure 1). Service replies with null.

6 /doodles/{id}/vote/{name}(GET)

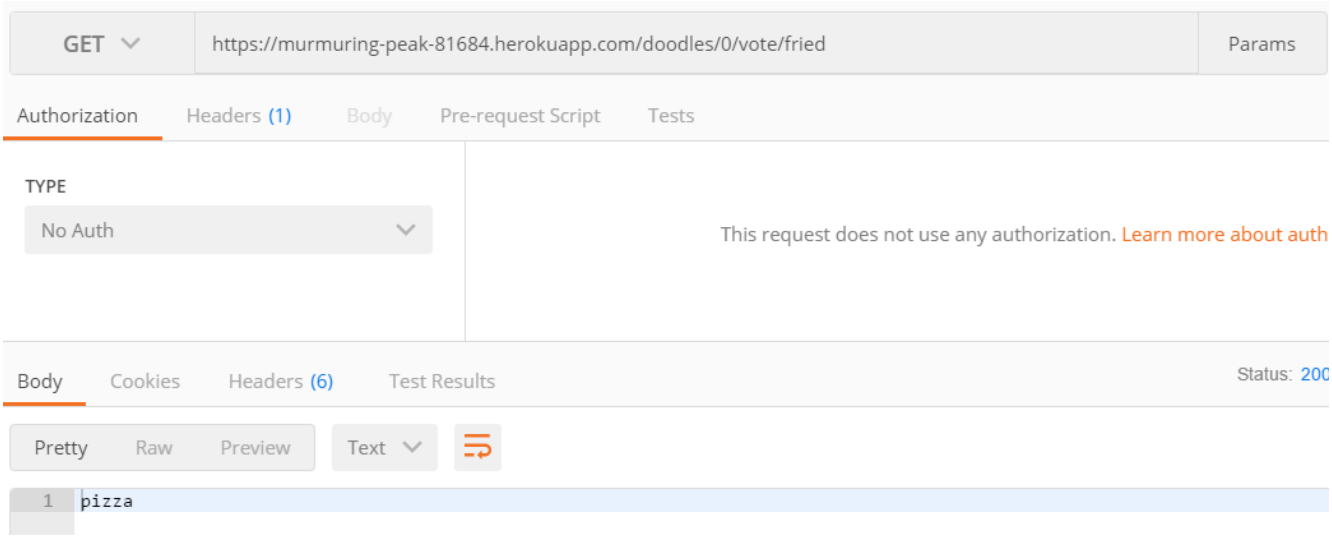

Figura 13: Correct GET request of the option voted by the user specified in the URI. Service replies with the option.

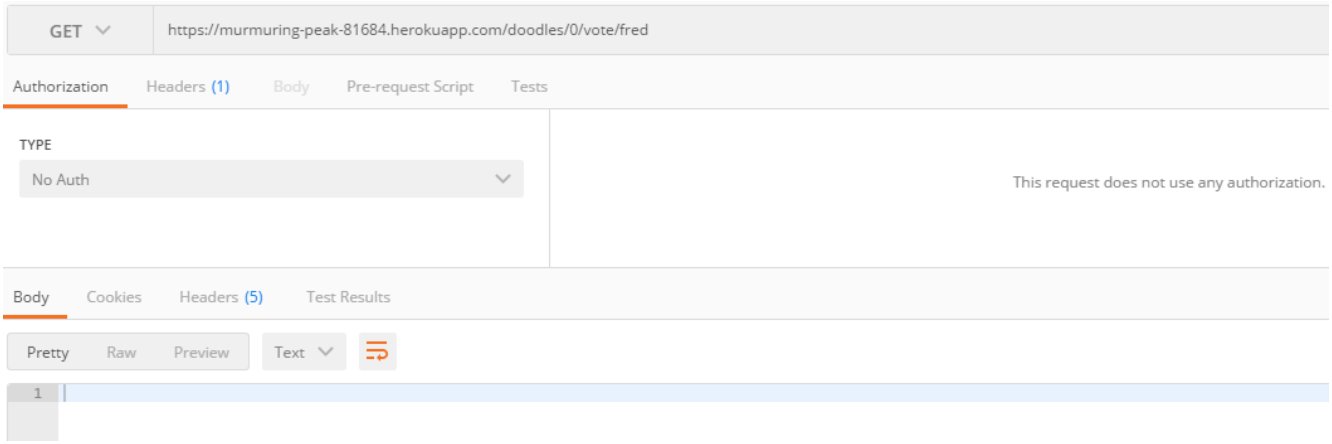

Figura 14: GET request of the option voted using a wrong name or inexistent name in the URI. Service replies with an empty message

7 /doodles/{id}/vote/{name}(POST)

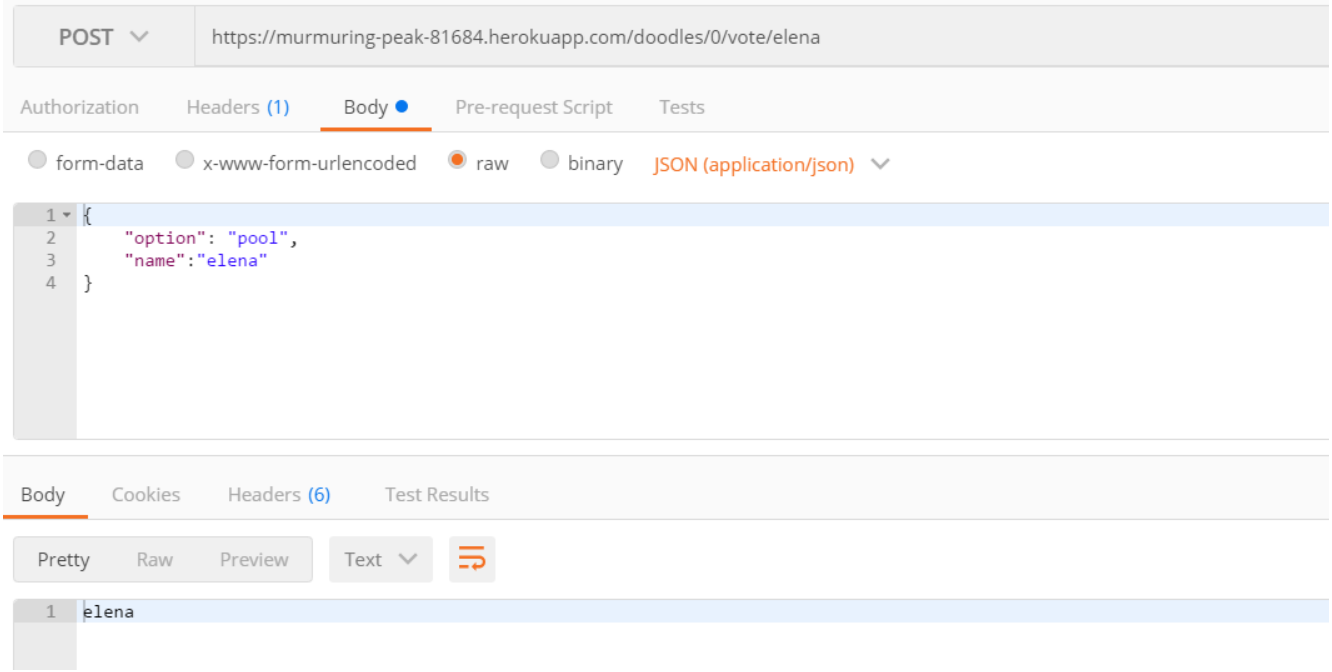

Figura 15: POST request to modify the vote of the user “elena”. The vote exists and the service replies with the voter’s name.

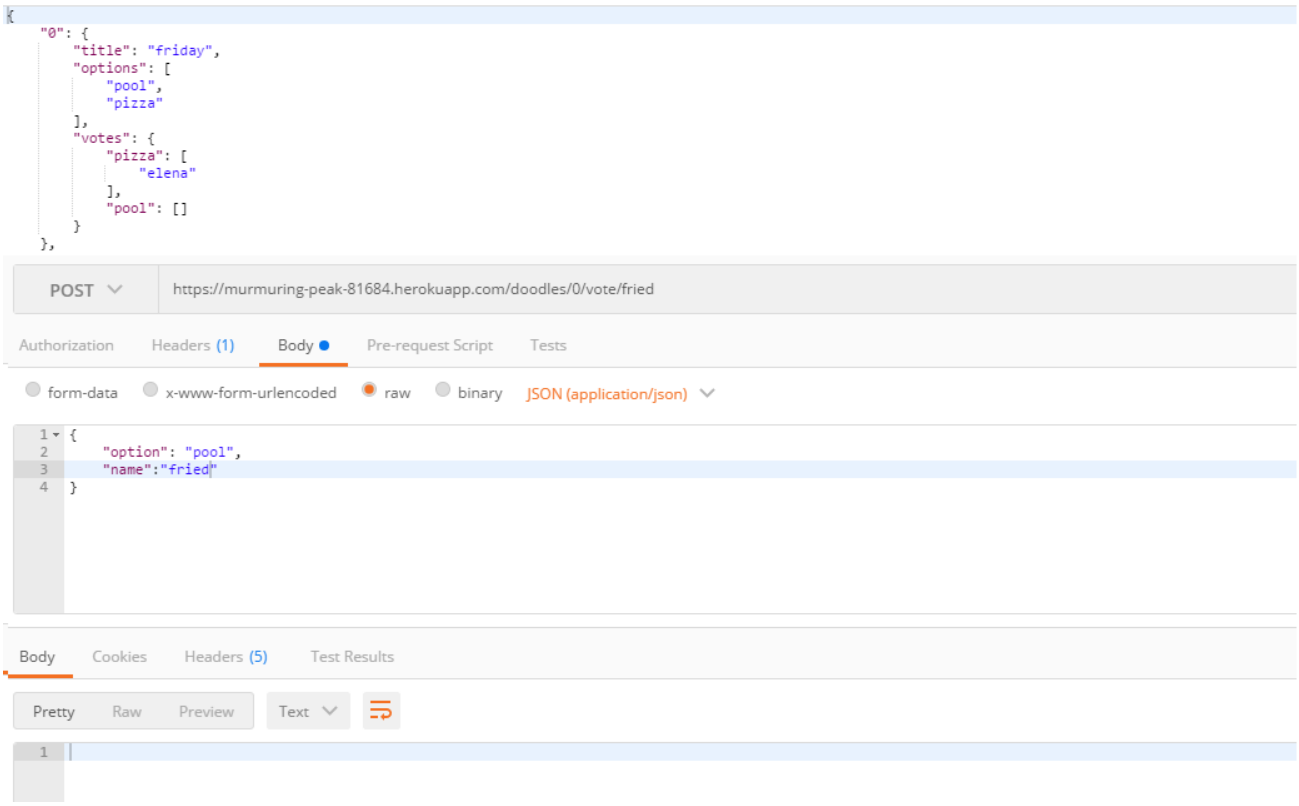

Figura 16: POST request to modify the vote of the user "fried". The vote does not exist and the service return null.

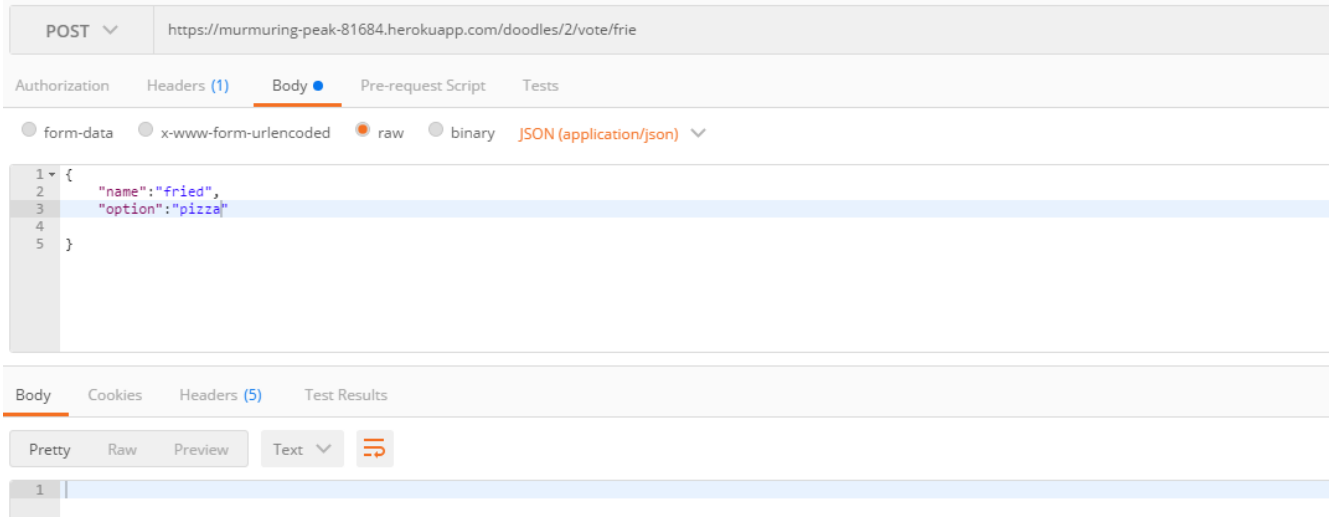

Figura 17: Wrong POST request to modify a vote. The name in the URI must be equal to the name in the body of message. Service returns null.

8 /doodles/{id}/vote/{name}(DELETE)

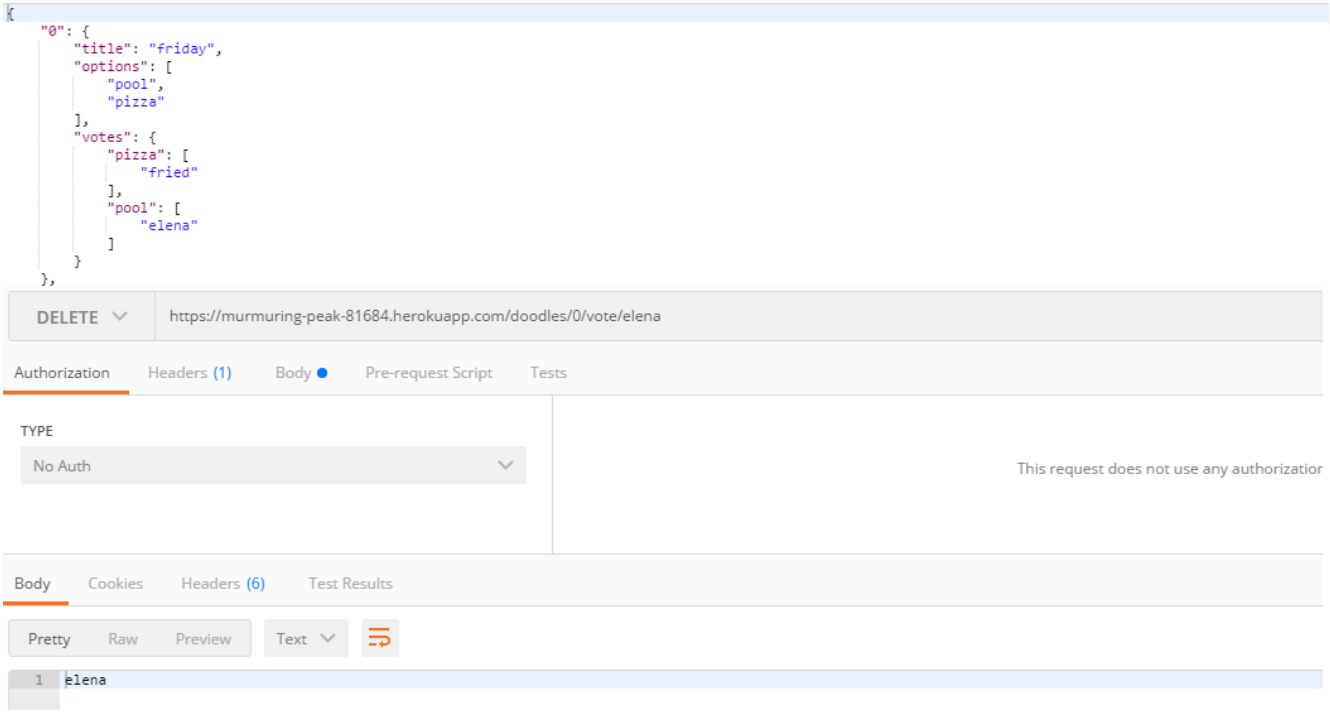

Figura 18: DELETE request to delete the vote of the user "elena". The vote exists, it is removed and the service returns the name.

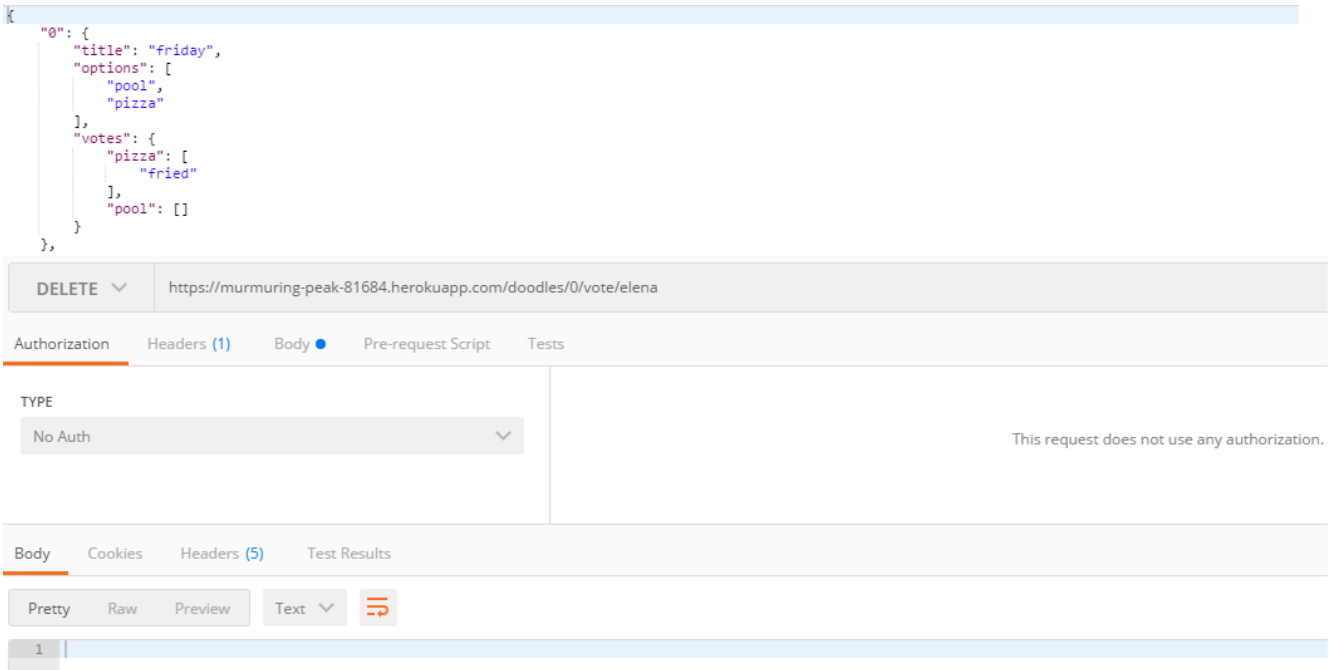

Figura 19: DELETE request to remove the vote of "elena". The name does not exists so service return null.