Homework 1

Francesco Carli - 559565

University of Pisa
M.Sc. Cybersecurity

Advanced Software Engineering course

1 Introduction

The homework implements a RESTful service with its API and it has been developed starting from a microservice skeleton. In Section 2 several screenshots have been inserted in order to show the successfull execution of the tests. Section 3 contains the link to the GitHub repository of the project.

2 Tests

Tests have been executed issuing the command *pytest* from the main project folder and the implemented solution passed all of them, as shown in Figure 1.

```
(venv) fcarli@fcarli3-HP-Laptop:~/Scaricati/skeleton/skeleton$ pytest --disable-warnings
========================================= test session starts ==========================================
platform linux -- Python 3.8.10, pytest-6.2.5, py-1.10.0, pluggy-1.0.0
rootdir: /home/fcarli/Scaricati/skeleton/skeleton, configfile: setup.cfg, testpaths: bedrock_a_party/tests
plugins: cov-3.0.0
collected 2 items

bedrock_a_party/tests/test_party.py ..                                                    [100%]

---------- coverage: platform linux, python 3.8.10-final-0 -----------
Name                               Stmts   Miss Branch BrPart  Cover   Missing
------------------------------------------------------------------------------
bedrock_a_party/classes/party.py      64     10     10      2    84%   17, 31, 39-40, 86, 94, 99, 102, 107, 110
bedrock_a_party/views/parties.py      74     10     22      5    82%   24->28, 51-54, 68->73, 95-96, 104-105, 123-124, 146
------------------------------------------------------------------------------
TOTAL                                189     20     32      7    87%

6 files skipped due to complete coverage.
Coverage HTML written to dir htmlcov

FAIL Required test coverage of 98% not reached. Total coverage: 86.88%

================================== 2 passed, 17 warnings in 0.83s ==================================
```

Figure 1: Output of pytest command.

2.1 Tests with PostMan

Test 1

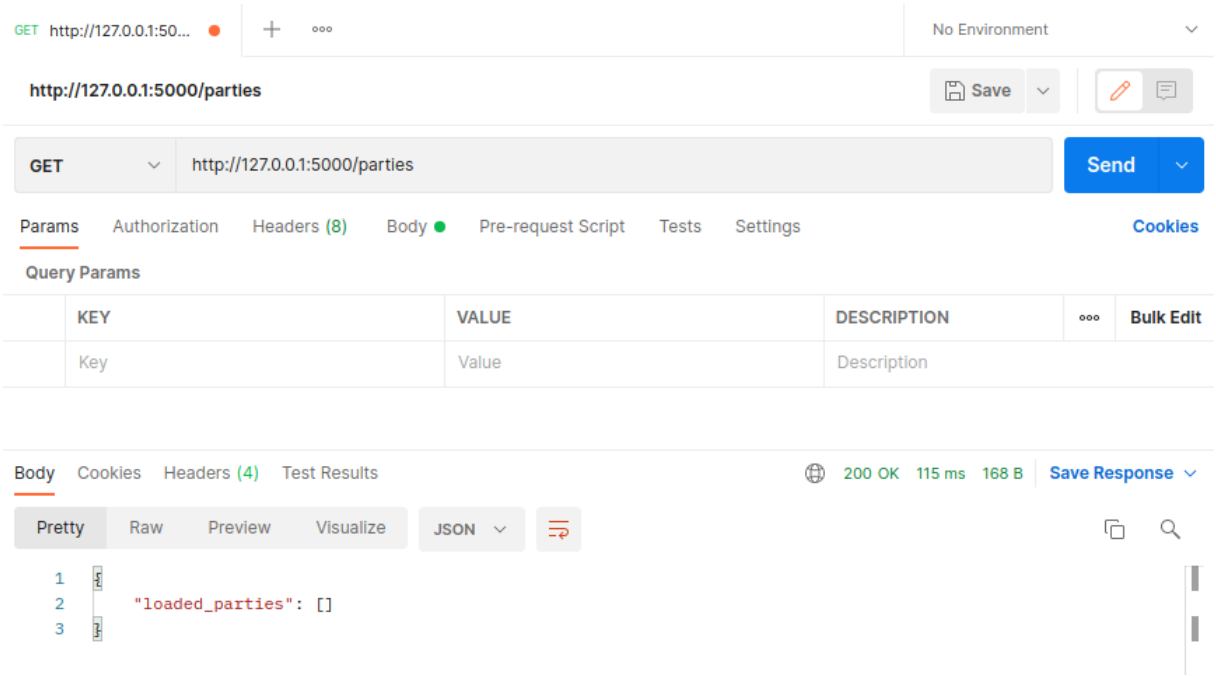

Figure 2

Figure 2 shows the output of a GET request on /parties with no existing parties yet.

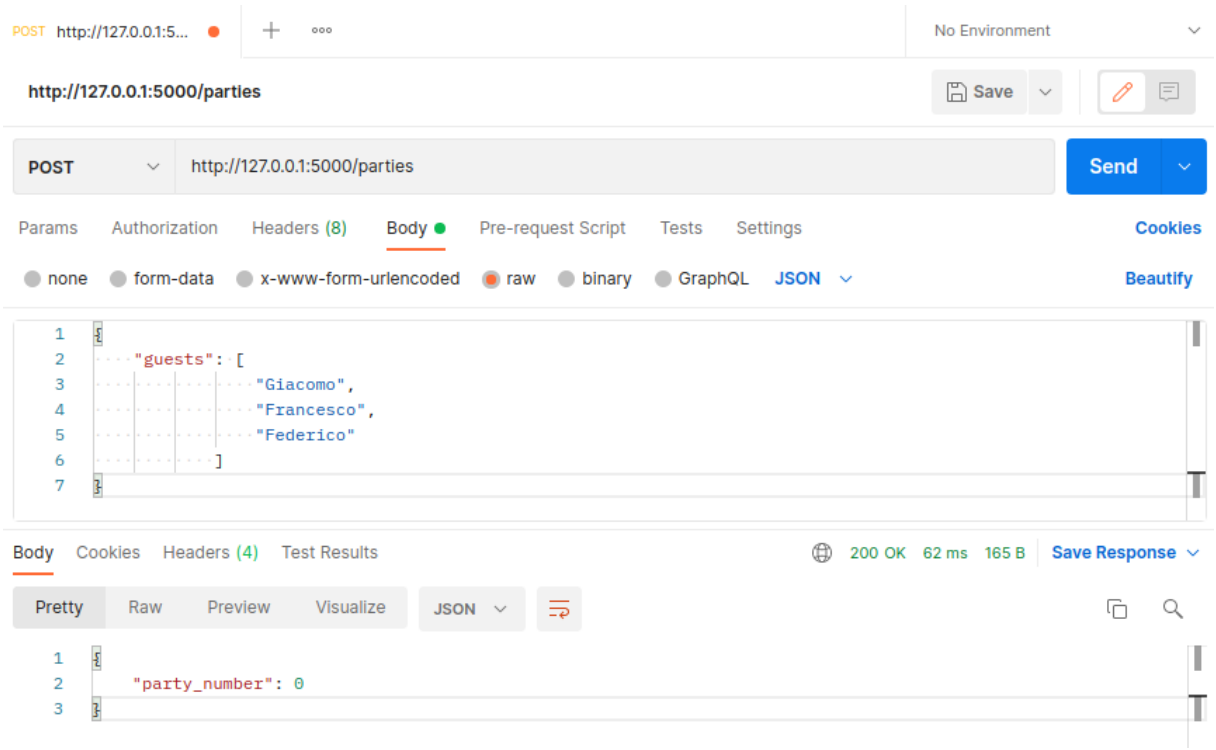

Figure 3

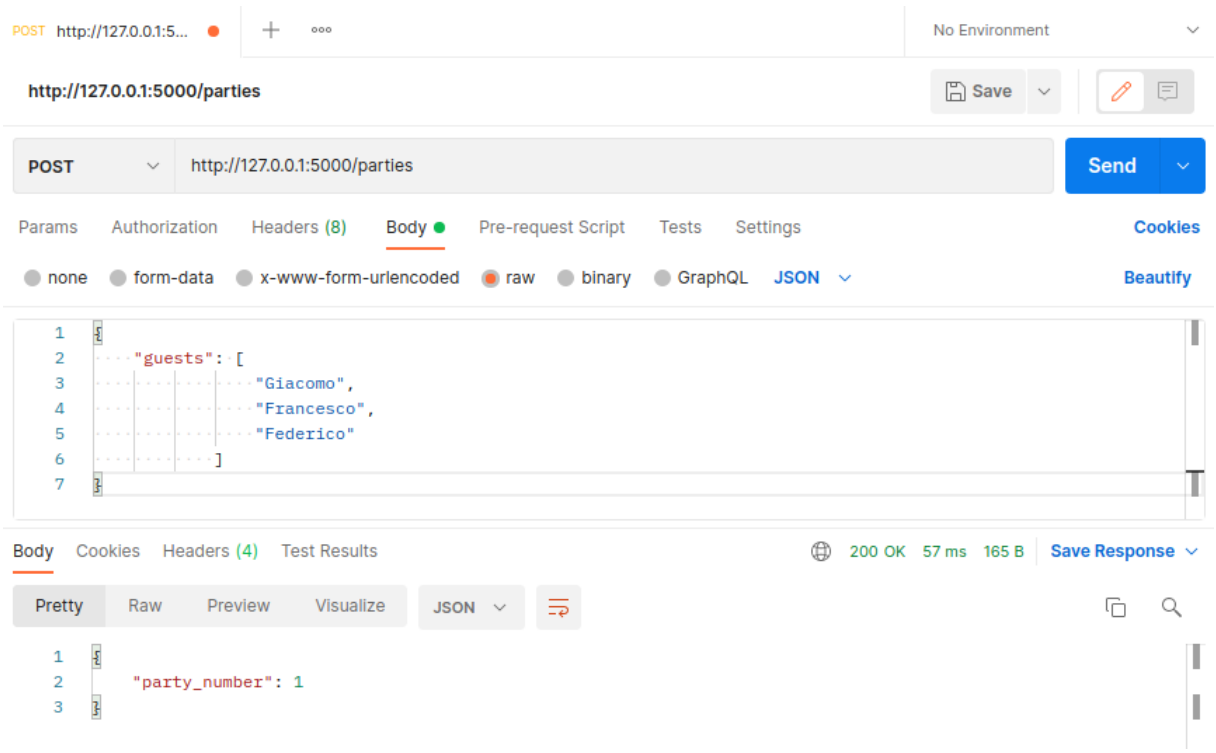

Figure 4

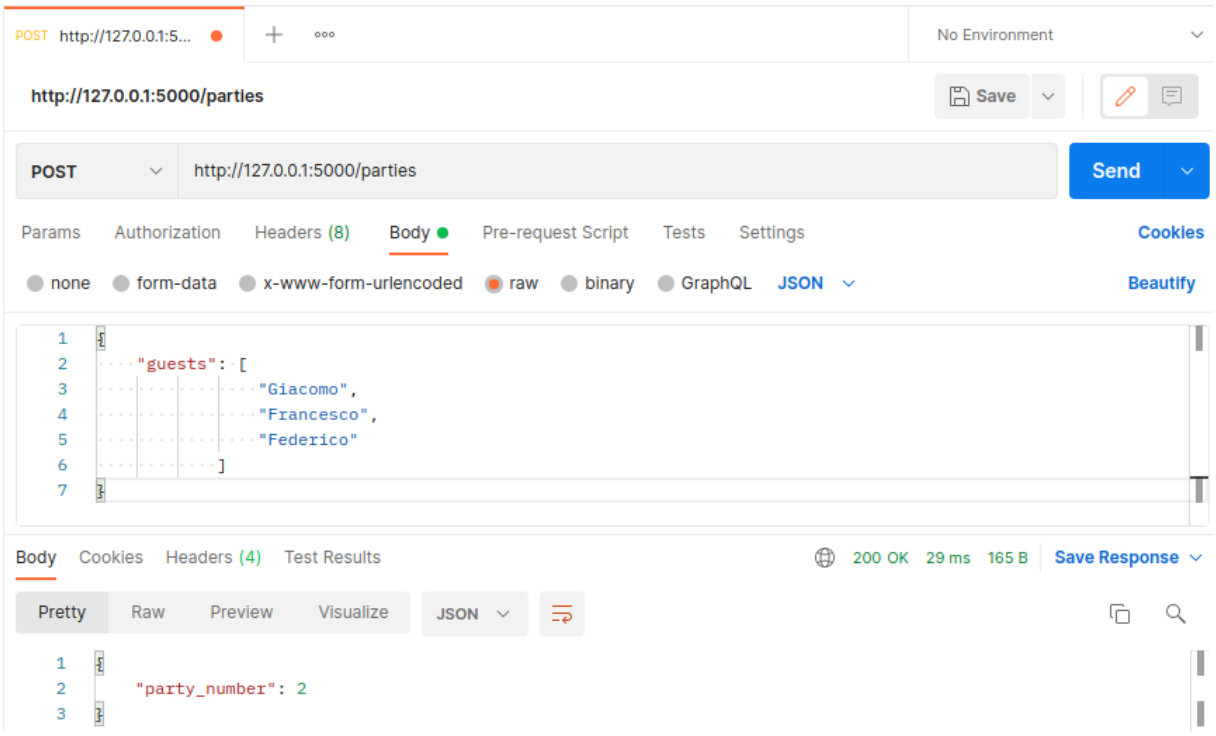

Figure 5

Figure 3, 4 and 5 show the outputs of POST requests on /parties. These requests creates three different parties.

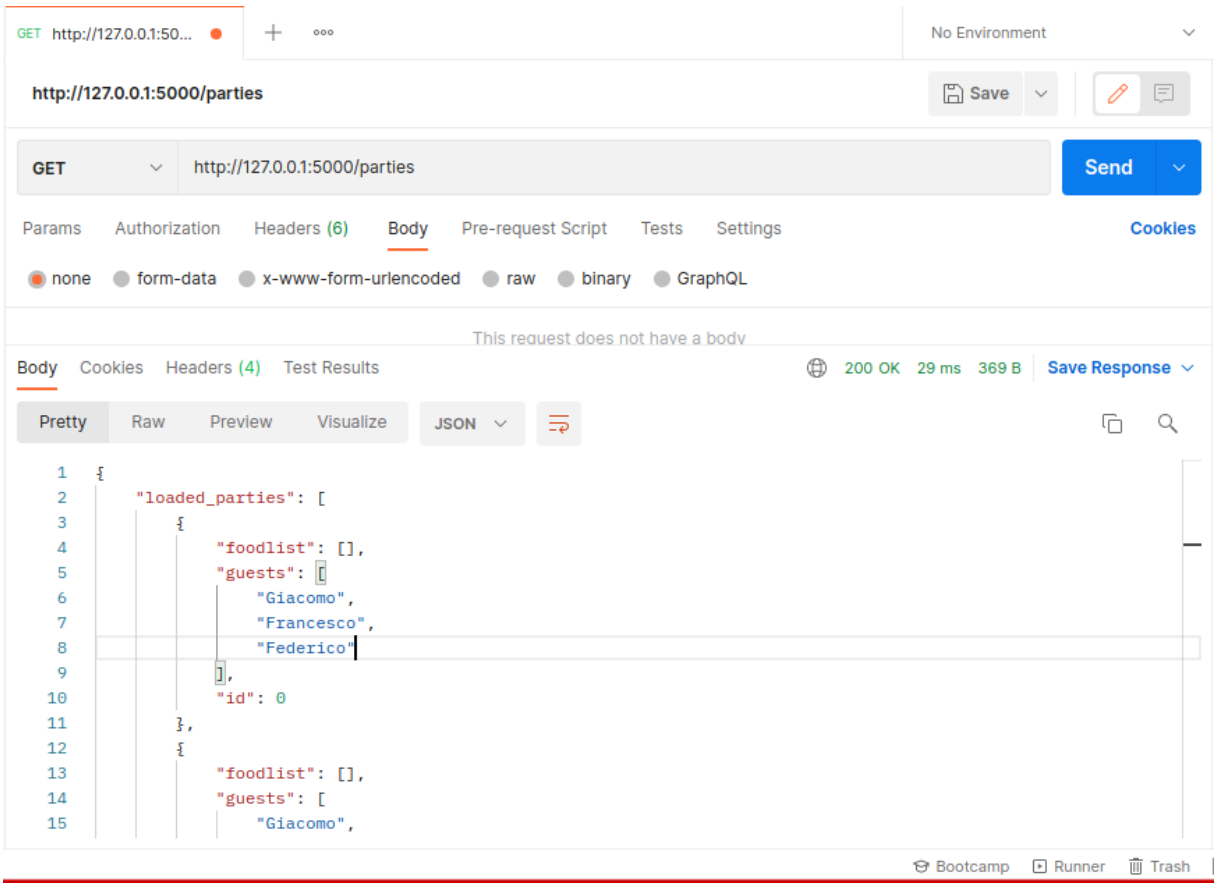

Figure 6

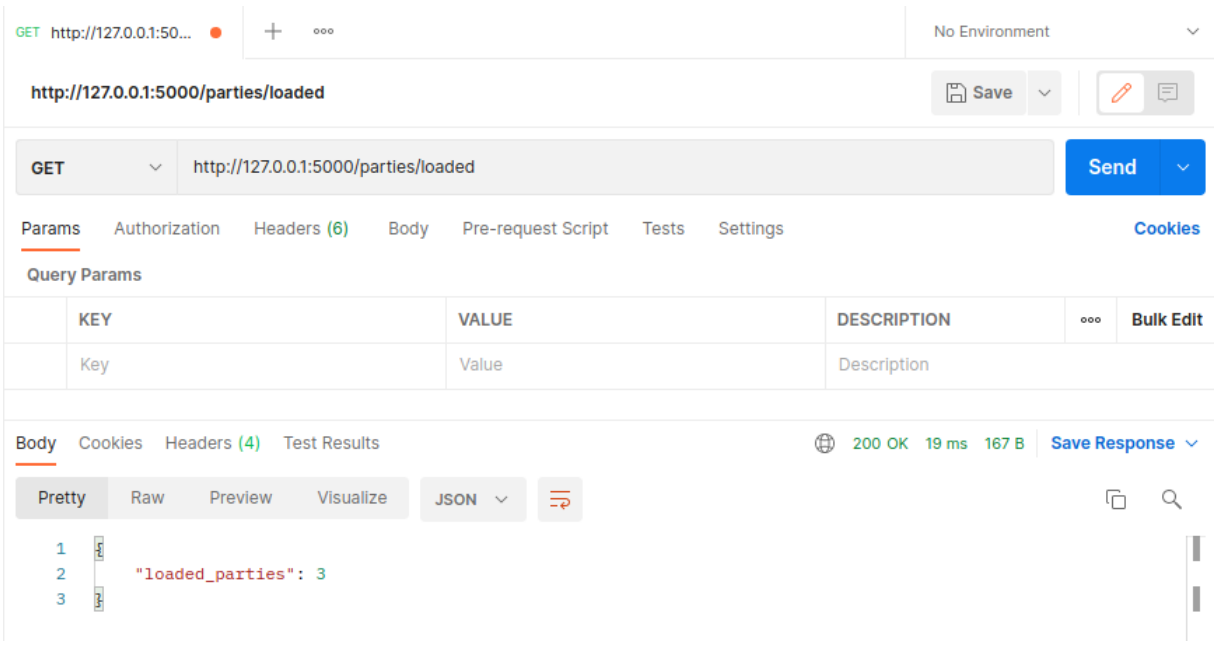

Figure 7

Figure 6 shows the partial output of a GET request on /parties, while Figure 7 shows the output of a GET request on /parties/loaded.

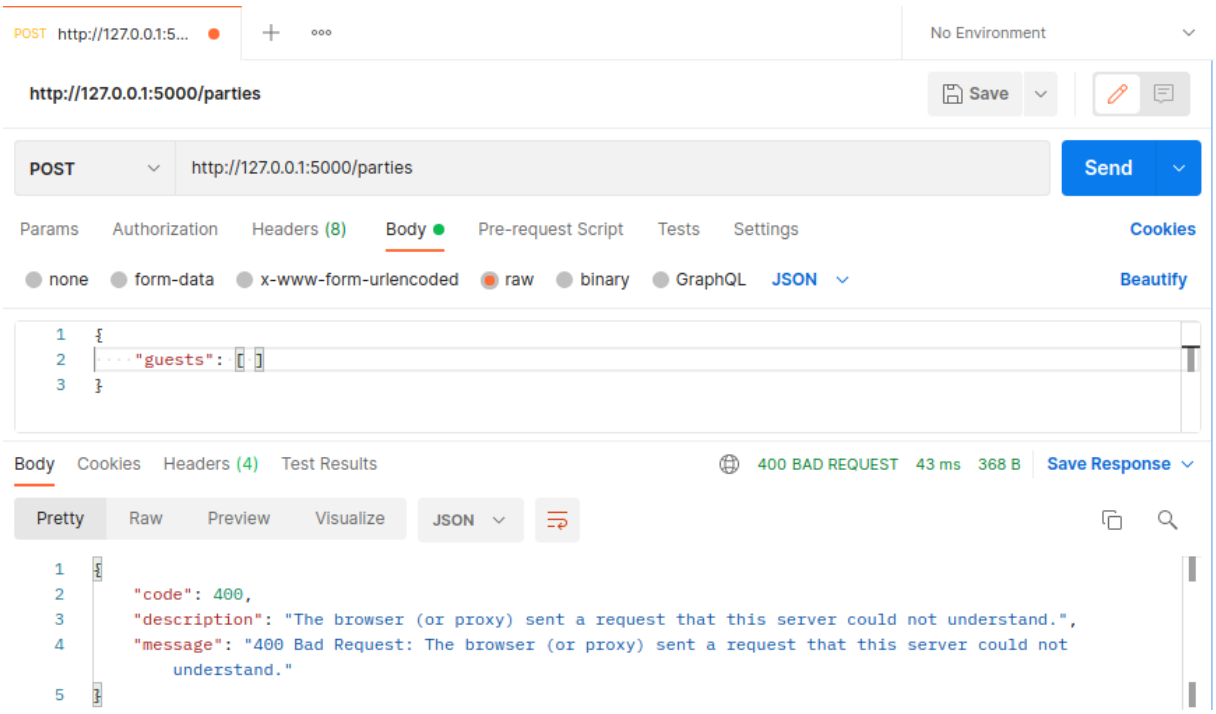

Figure 8

Figure 8 shows the output of a POST request on /parties with an empty list of guest. This request raises the *CannotPartyAloneError* exception.

Test 2

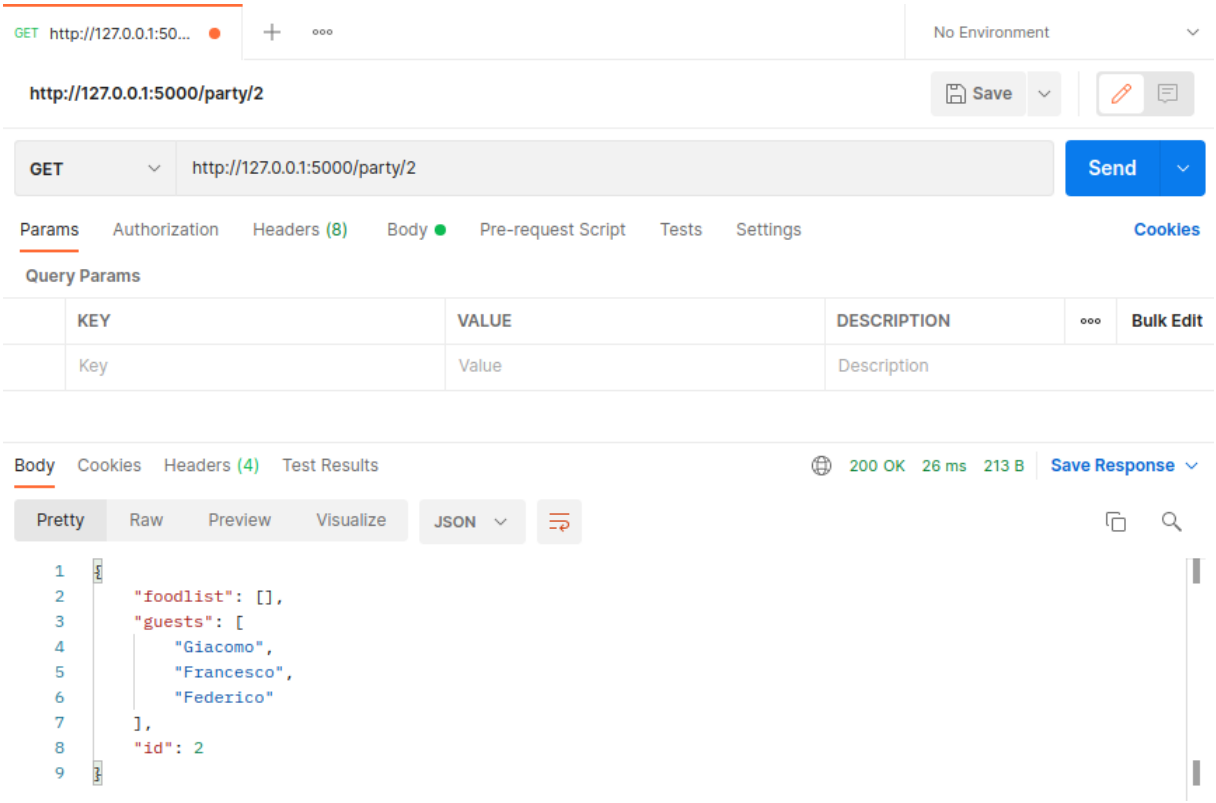

Figure 9

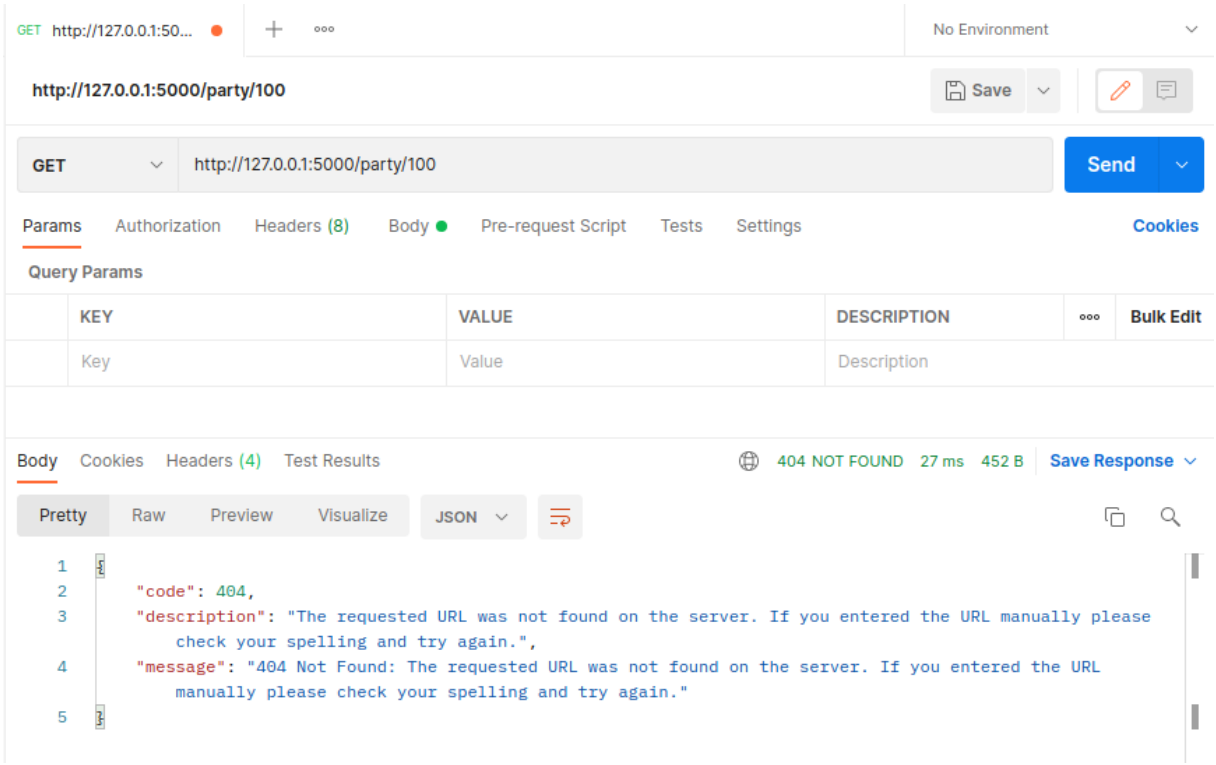

Figure 10

Figure 9 shows the output of a GET request on /party/2, (existing party). Figure 10 shows the output of a GET request on /party/100 (non-existing party).

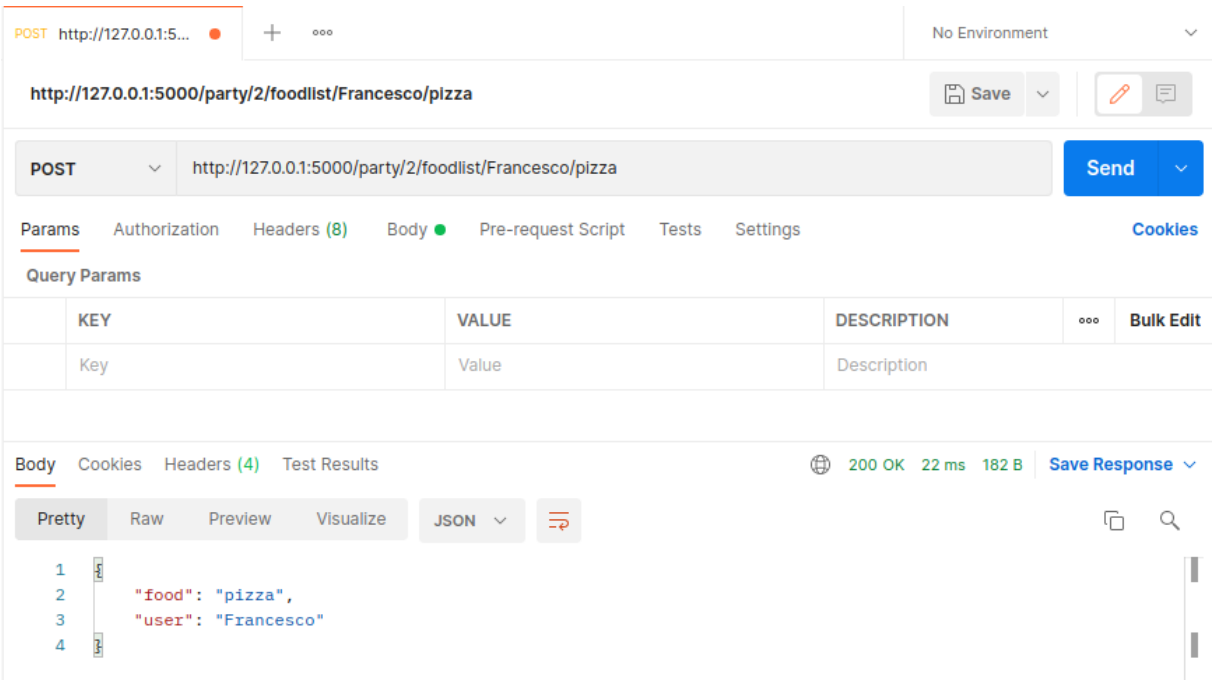

Figure 11

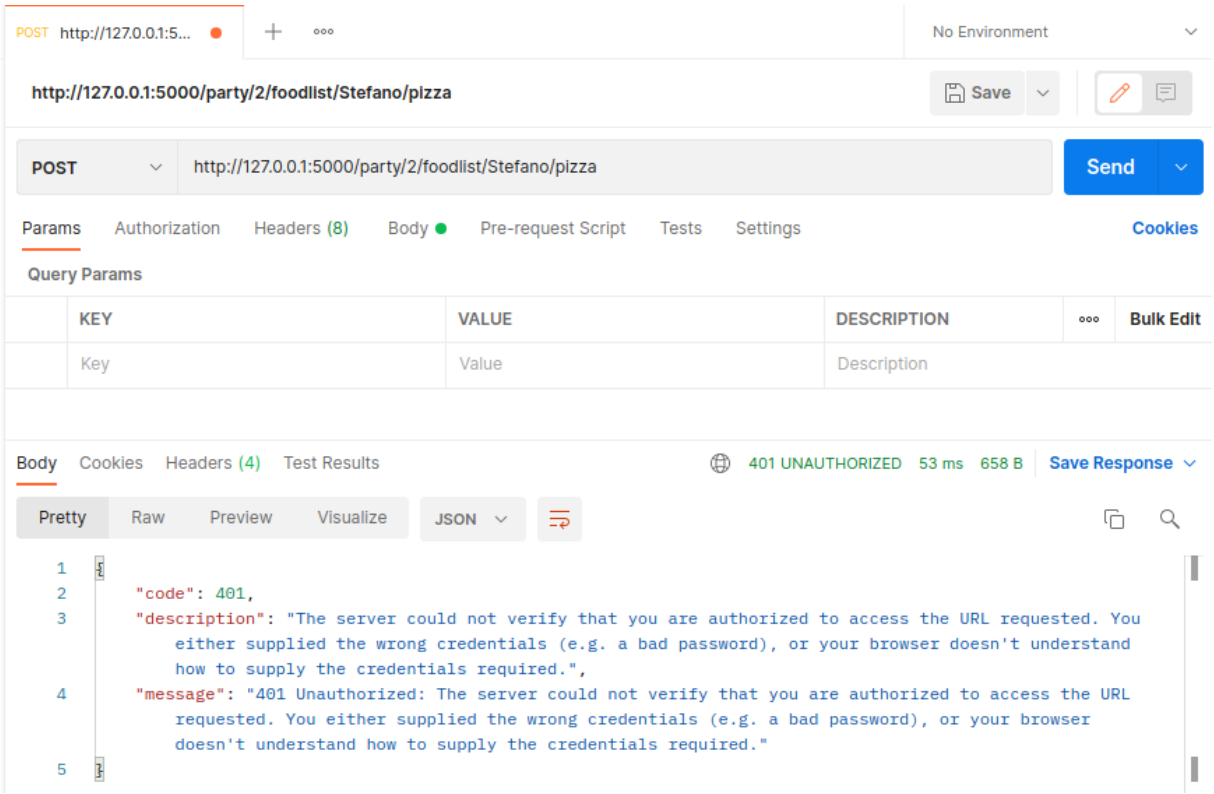

Figure 12

Figure 11 shows the output of a POST request on /party/2/foodlist/Francesco/pizza. Figure 12 shows a different output because the POST request is on /party/2/foodlist/Stefano/pizza, but Stefano is not a guest of the party, and this raises the *NotInvitedGuestError* exception.

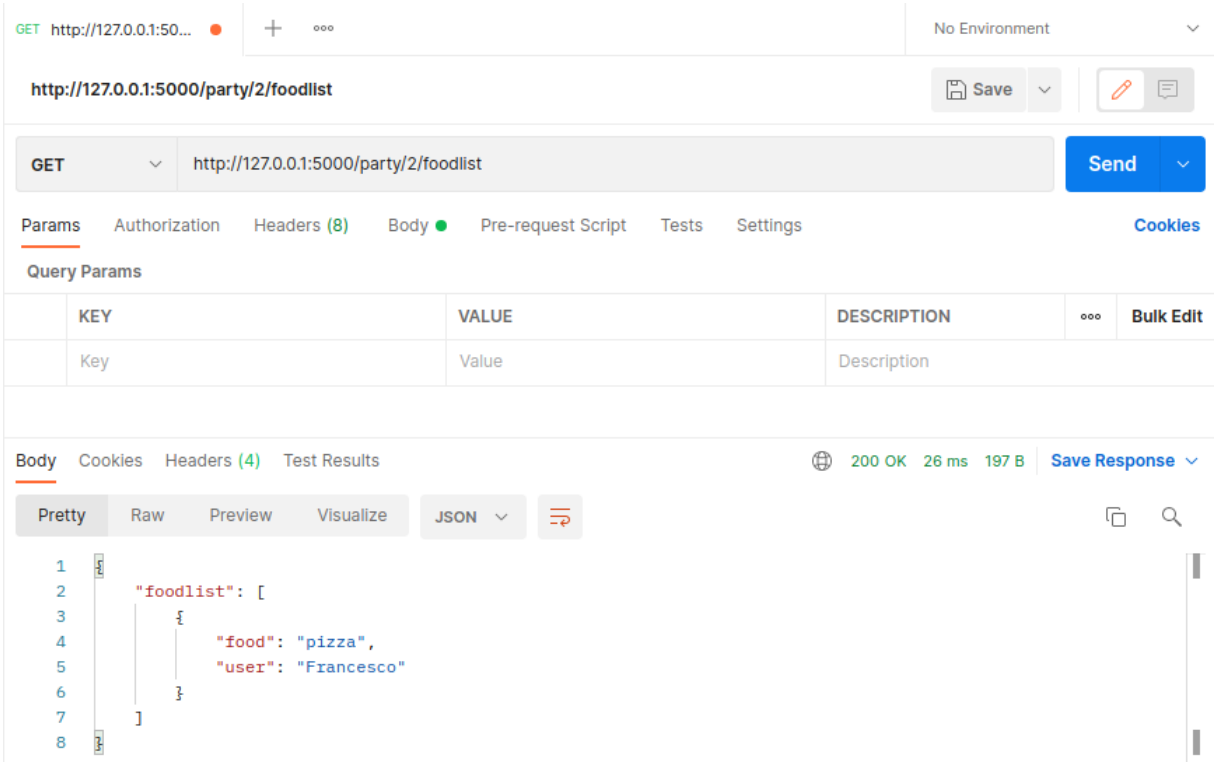

Figure 13

Figure 13 shows the output of a GET request on /party/2/foodlist.

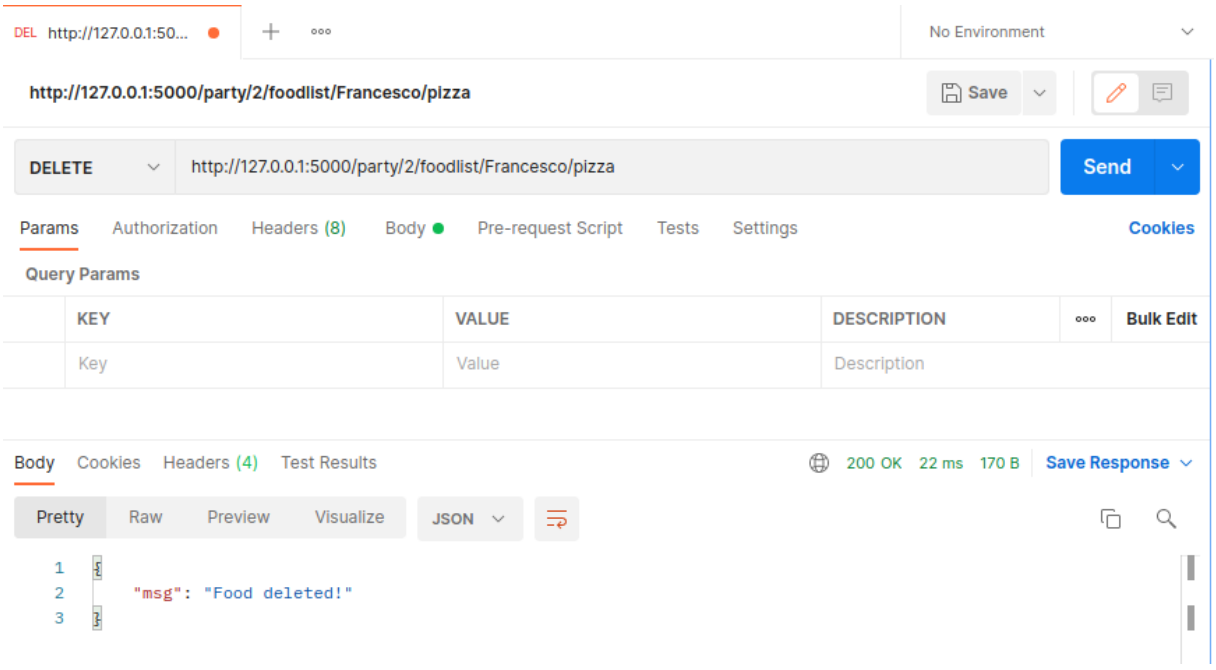

Figure 14

Figure 14 shows the output message of a DELETE request on /party/2/foodlist/Francesco/pizza. If the food to delete is not in the list, the *NotExistingFoodError* exception will be raised.

3 Reference

The entire homework is available on my GitHub page, at the following link: `https://github.com/fcarli3/advanced_software_engineering/tree/main/Homework1`