

## Report Assignment 1 Tommaso Lencioni 560309

### TODO LINK

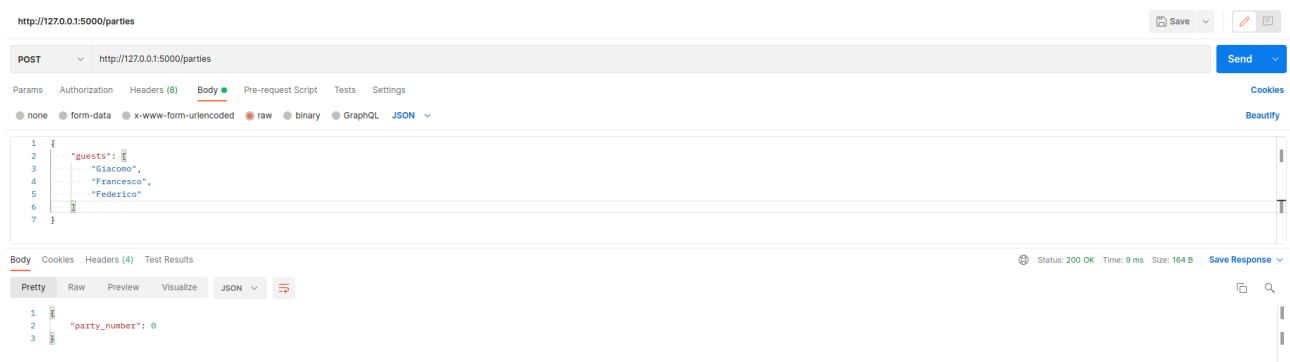
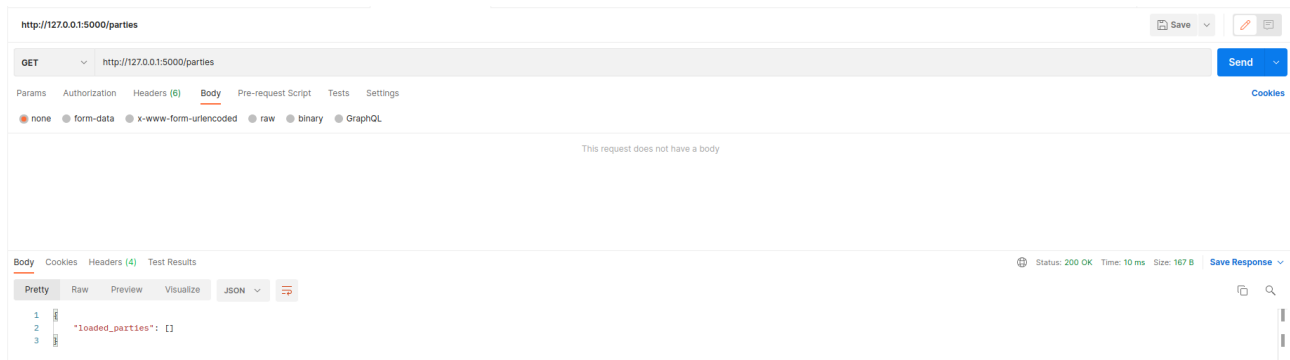
I followed the instruction given (hence creating a virtual environment and installing the modules in requirements.txt).

After completing the decorators with the correct HTTP methods allowed according to the table provided, I implemented all the methods.

(screenshots are small in order to catch all the information given by Postman)

### all\_parties:

- If the method is **POST** then a try-except block is opened and the provided method **create\_party** passing the request.
  - o If the content is bad formatted (no guests) an exception is raised and error code 400 is returned.
- If the method is **GET** the result of **get\_all\_parties** is returned (already jsonified by the provided function).



# Report Assignment 1 Tommaso Lencioni 560309

http://127.0.0.1:5000/parties

POST http://127.0.0.1:5000/parties

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "guests": [
3     "Giacomo",
4     "Francesco",
5     "Federico"
6   ]
7 }
```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 7 ms Size: 164 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "party_number": 1
3 }
```

http://127.0.0.1:5000/parties

POST http://127.0.0.1:5000/parties

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "guests": [
3     "Giacomo",
4     "Francesco",
5     "Federico"
6   ]
7 }
```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 16 ms Size: 164 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "party_number": 2
3 }
```

http://127.0.0.1:5000/parties

GET http://127.0.0.1:5000/parties

Params Authorization Headers (6) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 11 ms Size: 368 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "loaded_parties": [
3     {
4       "foodlist": [],
5       "guests": [
6         "Giacomo",
7         "Francesco",
8         "Federico"
9       ],
10      "id": 0
11    },
12    {
13      "foodlist": [],
14      "guests": [
15        "Giacomo",
16        "Francesco",
17        "Federico"
18      ],
19      "id": 1
20    },
21    {
22      "foodlist": [],
23      "guests": [
24        "Giacomo",
25        "Francesco",
```

## Report Assignment 1 Tommaso Lencioni 560309

The first screenshot shows a GET request to `http://127.0.0.1:5000/parties`. The response is a JSON array of three party objects, each with an `id`, a `foodlist`, and a `guests` array. The status is 200 OK.

```
[{"id": 0, "foodlist": [], "guests": ["Diacomo", "Francesco", "Federico"]}, {"id": 1, "foodlist": [], "guests": ["Diacomo", "Francesco", "Federico"]}, {"id": 2, "foodlist": [], "guests": ["Diacomo", "Francesco", "Federico"]}]
```

The second screenshot shows a POST request to `http://127.0.0.1:5000/parties`. The request body is a JSON object with a `guests` array. The response is a 400 BAD REQUEST, with the message "Empty party".

```
{ "guests": [] }
```

### loaded\_parties:

- Return the length of the dictionary containing the parties incapsulated in JSON with key **loaded\_parties**.

The screenshot shows a GET request to `http://127.0.0.1:5000/parties/loaded`. The response is a JSON object with a key `loaded_parties` and a value of 3. The status is 200 OK.

```
{ "loaded_parties": 3 }
```

### single\_party:

- Check whether the party exists.
- If the method is **GET** return the jsonified serialization of the party with **id** (taken from the URL).
- If the method is **DELETE** return the result of my utility function **delete\_party** (created for the modularity of the code) called with the **id** (taken from the URL).

## Report Assignment 1 Tommaso Lencioni 560309

The first screenshot shows a GET request to `http://127.0.0.1:5000/party/2`. The response status is 200 OK, and the body is a JSON object: `{ "foodlist": [], "guests": [ { "id": 1, "name": "Francesco", "surname": "Federico" }, { "id": 2, "name": "Francesco", "surname": "Federico" } ], "id": 2 }`.

The second screenshot shows a GET request to `http://127.0.0.1:5000/party/100`. The response status is 404 NOT FOUND, and the body is a JSON object: `{ "code": 404, "description": "The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.", "message": "404 Not Found: The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again." }`.

### get\_foodlist:

- Check whether the party exists.
- Return the serialization jsonified of the food list of the party **id** (taken from the URL).

The screenshot shows a GET request to `http://127.0.0.1:5000/party/2/foodlist`. The response status is 200 OK, and the body is a JSON object: `{ "foodlist": [ { "food": "pizza", "user": "Francesco" } ] }`.

### edit\_foodlist:

- Check whether the party exists.
- If the method is POST:
  - o In a try-except block **add\_to\_food\_list** is called with item and user taken from the URL, then the jsonification of them is returned.
  - o If the method calls raise an exception an according error code is returned instead.
- If the method is DELETE:
  - o In a try-except block **remove\_from\_food\_list** is called on the party **id** (taken from the URL) and return the jsonified message.
  - o If the food is not existing an error code (400) is returned instead.

# Report Assignment 1 Tommaso Lencioni 560309

http://127.0.0.1:5000/party/2/foodlist/Francesco/pizza

POST http://127.0.0.1:5000/party/2/foodlist/Francesco/pizza

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {"food": "pizza",
2  "user": "Francesco"
3 }
4
```

http://127.0.0.1:5000/party/2/foodlist/Stefano/pizza

POST http://127.0.0.1:5000/party/2/foodlist/Stefano/pizza

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize HTML

```
1 Not invited guest
```

http://127.0.0.1:5000/party/2/foodlist/Francesco/pizza

DELETE http://127.0.0.1:5000/party/2/foodlist/Francesco/pizza

Params Authorization Headers (6) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {"msg": "Food deleted!"
2 }
3
```

Both tests passed.

```
File Actions Edit View Help
venv [kali@kali]:~/Flask/HW1/ase-assignment1-master
python3
test session starts
platform linux -- Python 3.9.2, pytest-6.2.5, py-1.10.0, pluggy-1.0.0
rootdir: /home/kali/Flask/HW1/ase-assignment1-master, configFile: setup.cfg, testpaths: bedrock_a_party/tests
plugins: cov-3.0.0
collected 2 items
bedrock_a_party/tests/test_party.py ..

warnings summary
venv/lib/python3.9/site-packages/flask/json/__init__.py:31
/home/kali/Flask/HW1/ase-assignment1-master/venv/lib/python3.9/site-packages/flask/json/__init__.py:31: DeprecationWarning: Importing 'itsdangerous.json' is deprecated and will be removed in ItsDangerous 2.1. Use Python's 'json' module instead.
  _slash_escape = "\\" not in _json.dumps("/")
venv/lib/python3.9/site-packages/flask/json/__init__.py:61
/home/kali/Flask/HW1/ase-assignment1-master/venv/lib/python3.9/site-packages/flask/json/__init__.py:61: DeprecationWarning: Importing 'itsdangerous.json' is deprecated and will be removed in ItsDangerous 2.1. Use Python's 'json' module instead.
  class JSONEncoder(_json.JSONEncoder):
venv/lib/python3.9/site-packages/flask/json/__init__.py:103
/home/kali/Flask/HW1/ase-assignment1-master/venv/lib/python3.9/site-packages/flask/json/__init__.py:103: DeprecationWarning: Importing 'itsdangerous.json' is deprecated and will be removed in ItsDangerous 2.1. Use Python's 'json' module instead.
  class JSONDecoder(_json.JSONDecoder):
venv/lib/python3.9/site-packages/Flask/app.py:2459
/home/kali/Flask/HW1/ase-assignment1-master/venv/lib/python3.9/site-packages/Flask/app.py:2459: PytestCollectionWarning: cannot collect 'tested_app' because it is not a function.
  def __call__(self, environ, start_response):
bedrock_a_party/tests/test_party.py:11 warnings
/home/kali/Flask/HW1/ase-assignment1-master/venv/lib/python3.9/site-packages/flask/json/__init__.py:211: DeprecationWarning: Importing 'itsdangerous.json' is deprecated and will be removed in ItsDangerous 2.1. Use Python's 'json' module instead.
  rv = _json.dumps(obj, **kwargs)
-- Docs: https://docs.pytest.org/en/stable/warnings.html
----- coverage: platform linux, python 3.9.2-final-0 -----
Name                               Stmts   Miss Branch BrPart  Cover   Missing
bedrock_a_party/classes/party.py     64      10      18      2    84%   17, 31, 39-40, 86, 94, 99, 102, 107, 110
bedrock_a_party/views/parties.py     76      12      22      5    81%   21-23, 41-42, 52-54, 68-69, 75-76, 92-93, 115, 120-122
TOTAL                                191      22      32      7    86%
8 files skipped due to complete coverage.
Coverage HTML written to dir htmlcov
FAIL Required test coverage of 90% not reached. Total coverage: 86.10%
2 passed, 15 warnings in 0.59s
```