

**Advanced Software Engineering**  
University of Pisa  
Homework 1  
Daniele Gadler  
564198

1. Link to the Github Repository:

<https://github.com/DanyEle/ase-lab-18/tree/master/hw-1>

2. Screenshot of successful execution of myservice/tests/test\_doodle.py

```
dantelegd@ant-ubuntu:~/PycharmProjects/doodle/myservice$ pytest
platform linux -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0
rootdir: /home/dantelegd/PycharmProjects/doodle, inifile:
plugins: remotestests-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, arraydiff-0.2
collected 3 items

test_doodle.py ...

myservice/tests/test_doodle.py: Cannot collect 'tested_app' because it is not a function.

-- Docs: http://doc.pytest.org/en/latest/warnings.html
myservice/tests/test_doodle.py: 3 passed, 1 warnings in 0.22 seconds
```

All tests have been passed, even though one warning has been generated.

**Further details:**

In my code, I added a few more exceptions, throwing (400) client errors in case the input provided does not fit the foreseen type (for example, if a string is expected, but something different from a string is passed).

3. Screenshots of the tests performed with Postman for all above operations:

**[Test 1 – Create Doodle test and get doodle]**

**1) Create doodle**

POST - http://127.0.0.1:5000/doodles

Body

```
{
  "title": "Hello",
  "options": [
    "1",
    "2",
    "3"
  ]
}
```

Body

```
{
  "id": 1
}
```

2)

The screenshot shows the Postman application interface. In the left sidebar, under 'Collections', there is a section for 'Doodles' with 7 requests. The main workspace displays a POST request to 'http://127.0.0.1:5000/doodles'. The 'Body' tab is selected, showing the following JSON payload:

```
1 - {  
2   "title": "pollz",  
3   "options": ["1", "2"]  
4 }
```

Below the body, the response status is 200 OK, time 3 ms, and size 163 B. The response body is shown in JSON format:

```
1 - [ {  
2   "pollnumber": 2  
3 }]
```

3)

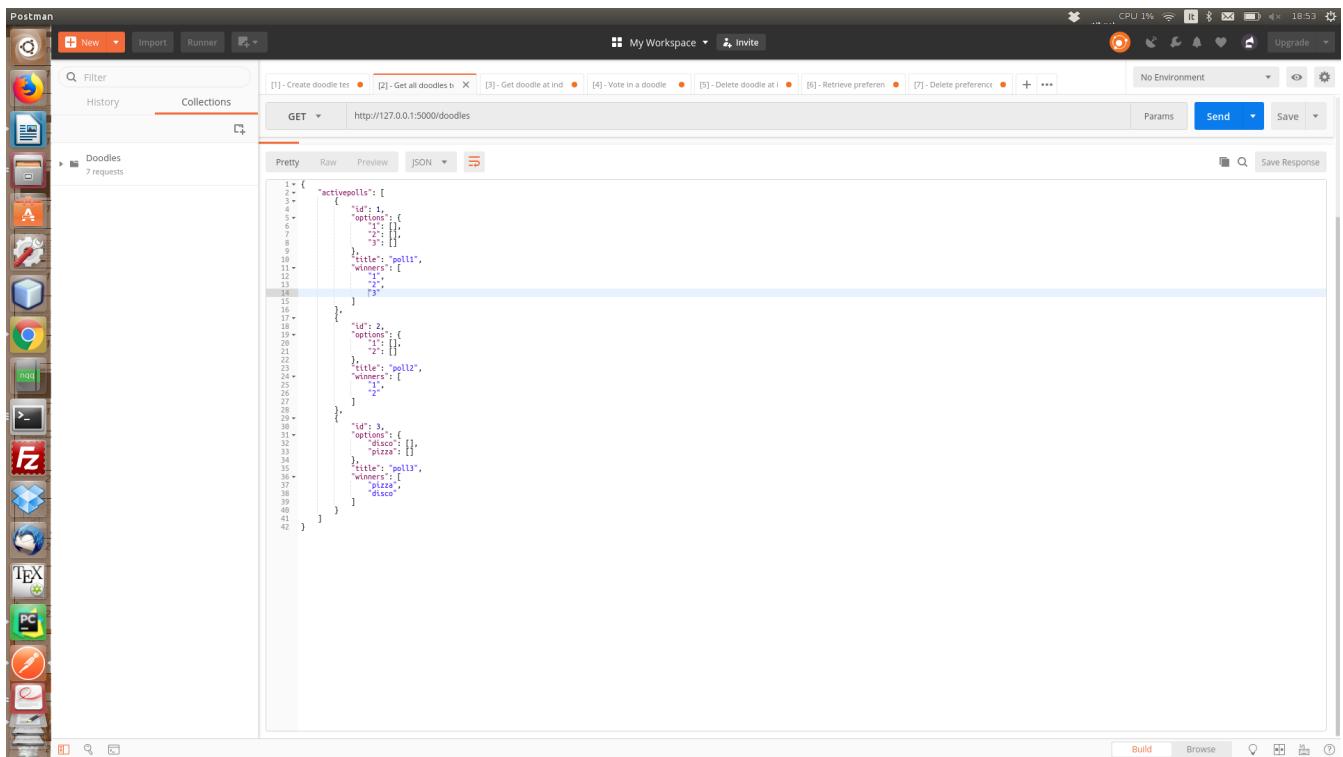
The screenshot shows the Postman application interface. In the left sidebar, under 'Collections', there is a section for 'Doodles' with 7 requests. The main workspace displays a POST request to 'http://127.0.0.1:5000/doodles'. The 'Body' tab is selected, showing the following JSON payload:

```
1 - {  
2   "title": "pollz",  
3   "options": ["mizza", "disco"]  
4 }
```

Below the body, the response status is 200 OK, time 11 ms, and size 163 B. The response body is shown in JSON format:

```
1 - [ {  
2   "pollnumber": 3  
3 }]
```

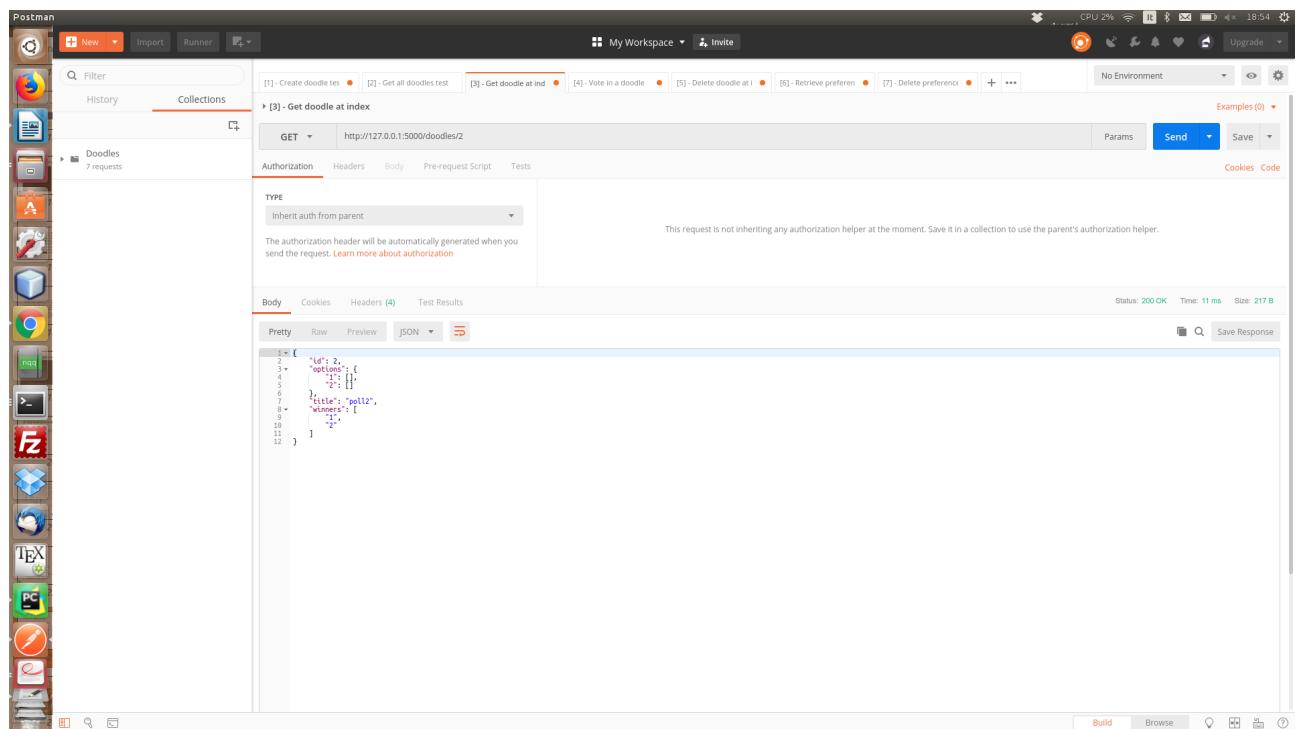
#### 4) Get all doodles



```
[{"id": 1, "options": [{"id": 1, "option": "1"}, {"id": 2, "option": "2"}, {"id": 3, "option": "3"}], "title": "poll1", "winner": ["1", "2", "3"]}, {"id": 2, "options": [{"id": 1, "option": "1"}, {"id": 2, "option": "2"}], "title": "poll2", "winner": ["1", "2"]}, {"id": 3, "options": [{"id": 1, "option": "pizza"}, {"id": 2, "option": "disco"}], "title": "poll3", "winner": ["pizza", "disco"]}],
```

#### [Test 2 – retrieve single doodle]

#### 5) Get a single doodle



```
{"id": 2, "options": [{"id": 1, "option": "1"}, {"id": 2, "option": "2"}], "title": "poll2", "winner": ["1", "2"]}]
```

6)

The screenshot shows the Postman application interface. A collection named "Doodles" is selected. A request titled "[3] - Get doodle at index" is being viewed. The method is set to GET, and the URL is http://127.0.0.1:5000/doodles/12. The "Authorization" tab is active, showing "Inherit auth from parent". The response status is 404 NOT FOUND, with a message: "The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again." Below the response, the JSON body is displayed:

```
[{"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."}]
```

7) Test vote in a doodle

The screenshot shows the Postman application interface. A collection named "Doodles" is selected. A request titled "[4] - Vote in a doodle" is being viewed. The method is set to PUT, and the URL is http://127.0.0.1:5000/doodles/2. The "Body" tab is active, showing the JSON body:

```
{"person": "fred", "option": "1"}
```

The response status is 200 OK, with a message: "{'winners': [1]}". Below the response, the JSON body is displayed:

```
{ "winners": [1] }
```

8)

The screenshot shows the Postman application interface. A PUT request is being made to the URL `http://127.0.0.1:5000/doodles/1`. The request body is set to `JSON (application/json)` and contains the following JSON payload:

```
[{"person": "fred", "option": "2"}]
```

The response status is `200 OK`, time is `9 ms`, and size is `168 B`. The response body is displayed in JSON format:

```
{"winners": [{"id": 1, "name": "fred"}]}
```

9)

The screenshot shows the Postman application interface. A PUT request is being made to the URL `http://127.0.0.1:5000/doodles/2`. The request body is set to `JSON (application/json)` and contains the following JSON payload:

```
[{"person": "fred", "option": "2"}]
```

The response status is `200 OK`, time is `9 ms`, and size is `168 B`. The response body is displayed in JSON format:

```
{"winners": [{"id": 1, "name": "fred"}]}
```

10)

The screenshot shows the Postman application interface. In the center, there is a request panel titled 'PUT' with the URL 'http://127.0.0.1:5000/doodles/2'. The 'Body' tab is selected, showing a JSON payload:

```
{ "person": "barney", "option": "I" }
```

Below the request, the response section displays a status of '200 OK' with a time of '10 ms' and a size of '164 B'. The response body is shown in JSON format:

```
{ "votes": [ "I" ] }
```

## 11) Same person voting already voted option from themselves

The screenshot shows the Postman application interface. In the center, there is a request panel titled 'PUT' with the URL 'http://127.0.0.1:5000/doodles/2'. The 'Body' tab is selected, showing a JSON payload:

```
{ "person": "barney", "option": "I" }
```

Below the request, the response section displays a status of '400 BAD REQUEST' with a time of '8 ms' and a size of '368 B'. The response body is shown in JSON format:

```
{ "code": 400, "description": "The browser (or proxy) sent a request that this server could not understand.", "message": "400 Bad Request: The browser (or proxy) sent a request that this server could not understand." }
```

## 12) Vote non-existing option

The screenshot shows the Postman application interface. A PUT request is being made to `http://127.0.0.1:5000/doodles/2`. The request body is set to `JSON (application/json)` and contains the following JSON:

```
1 - {  
2 -   "person": "wimla",  
3 -   "option": "g"  
4 - }
```

The response status is `400 BAD REQUEST` with a message: `"code": 400, "description": "The browser (or proxy) sent a request that this server could not understand.", "message": "400 Bad Request: The browser (or proxy) sent a request that this server could not understand."`.

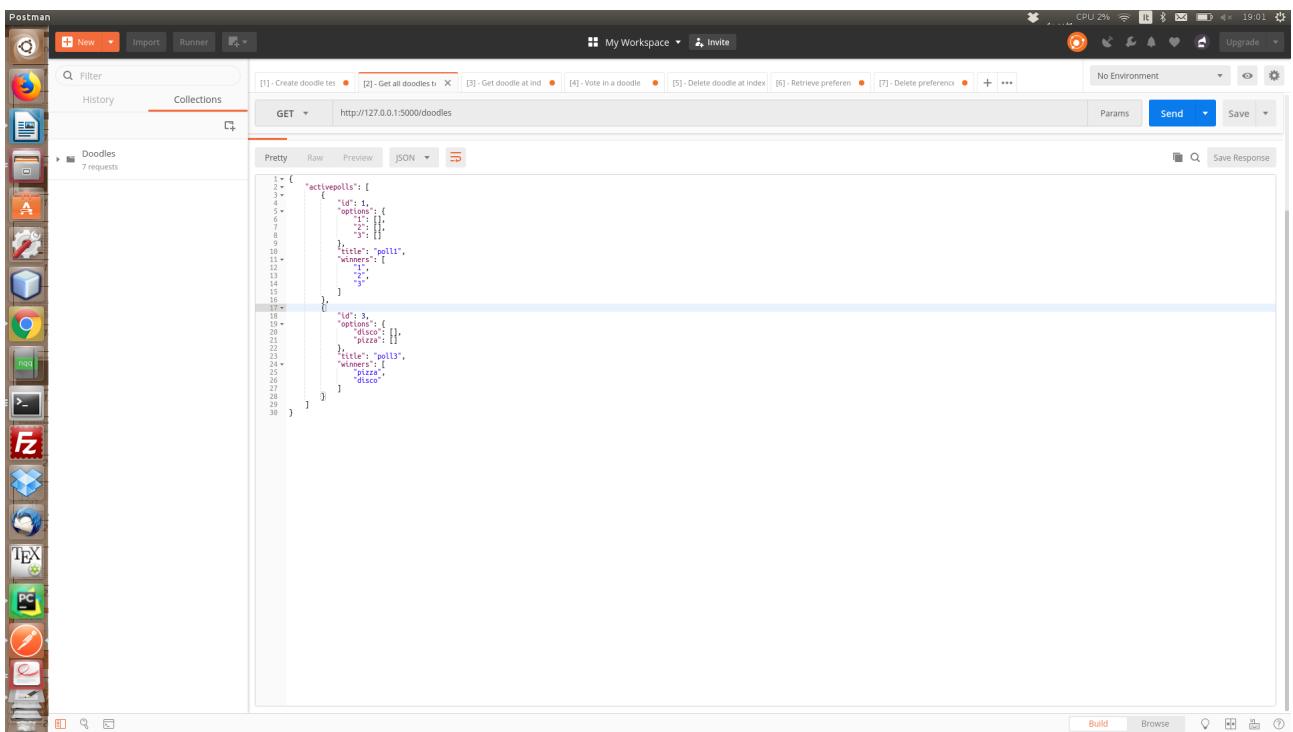
### 13) Delete a doodle

The screenshot shows the Postman application interface. A DELETE request is being made to `http://127.0.0.1:5000/doodles/2`. The request body is set to `JSON (application/json)` and contains the following JSON:

```
1 - {  
2 -   "wimla": [  
3 -     "1"  
4 -   ]  
5 - }
```

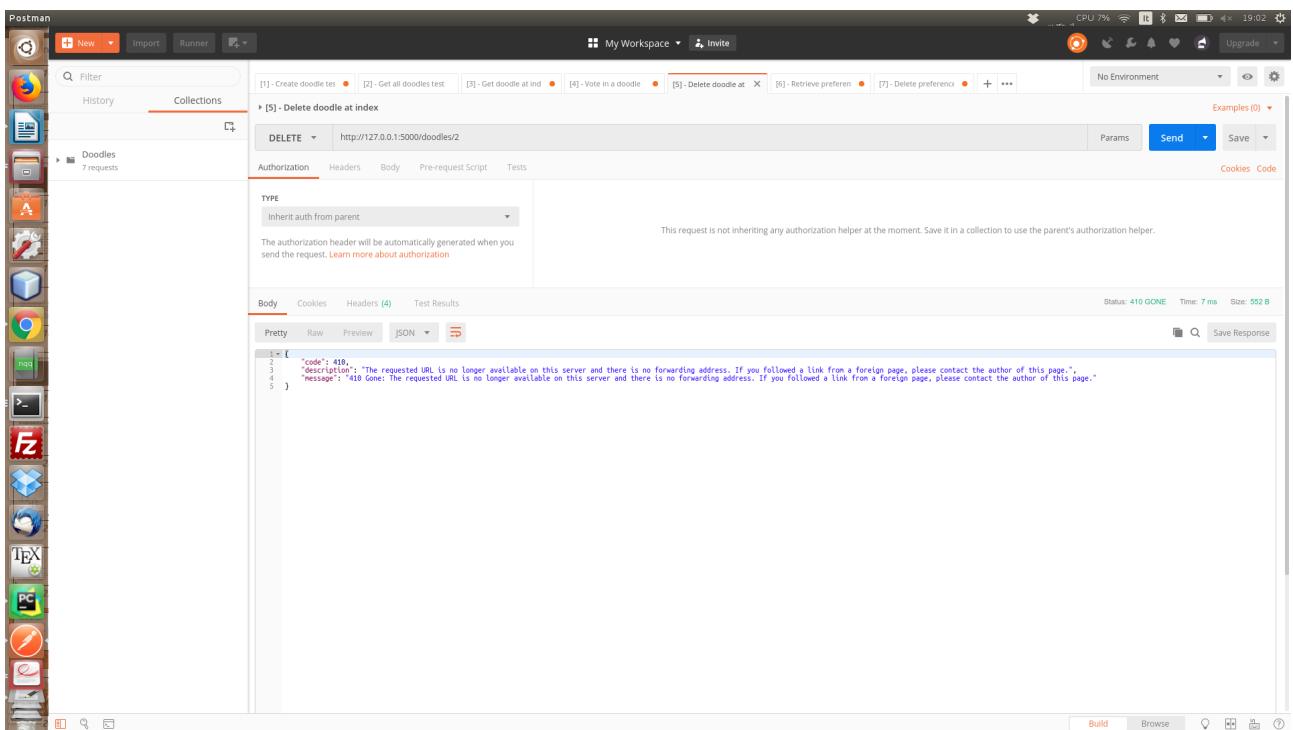
The response status is `200 OK`.

### 14) Get all existing doodles



```
[{"id": 1, "title": "polli", "winners": ["polli"], "options": [{"id": 1, "text": "polli"}, {"id": 2, "text": "disco"}]}, {"id": 3, "title": "polli", "winners": ["disco"], "options": [{"id": 1, "text": "polli"}, {"id": 2, "text": "disco"}]}]
```

## 15) Delete previous doodle



```
{"code": 410, "description": "The requested URL is no longer available on this server and there is no forwarding address. If you followed a link from a foreign page, please contact the author of this page.", "message": "410 Gone: The requested URL is no longer available on this server and there is no forwarding address. If you followed a link from a foreign page, please contact the author of this page."}
```

## 16) Delete non-existing doodle

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for 'New', 'Import', 'Runner', and 'Collections'. The 'Collections' tab is selected. Below the tabs, there is a search bar labeled 'Filter' and a dropdown menu for 'History' and 'Collections'. A sidebar on the left contains icons for various tools and services, including a browser icon, a file icon, and a database icon. The main workspace shows a collection named 'Doodles' with 7 requests. A specific request is selected for deletion, indicated by a red border around the 'DELETE' button and the URL 'http://127.0.0.1:5000/doodles/12'. The 'Authorization' tab is active, showing 'Inherit auth from parent'. The response body is displayed in JSON format, showing an error message: { "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." }. The status bar at the bottom indicates 'Status: 404 NOT FOUND'.

## 17) Get previously existing doodle

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for 'New', 'Import', 'Runner', and 'Collections'. The 'Collections' tab is selected. Below the tabs, there is a search bar labeled 'Filter' and a dropdown menu for 'History' and 'Collections'. A sidebar on the left contains icons for various tools and services, including a browser icon, a file icon, and a database icon. The main workspace shows a collection named 'Doodles' with 7 requests. A specific request is selected for retrieval, indicated by a red border around the 'GET' button and the URL 'http://127.0.0.1:5000/doodles/2'. The 'Authorization' tab is active, showing 'Inherit auth from parent'. The response body is displayed in JSON format, showing an error message: { "code": 410, "description": "The requested URL is no longer available on this server and there is no forwarding address. If you followed a link from a foreign page, please contact the author of this page.", "message": "410 Gone: The requested URL is no longer available on this server and there is no forwarding address. If you followed a link from a foreign page, please contact the author of this page." }. The status bar at the bottom indicates 'Status: 410 GONE'.

[Test 3 ]

## 18) Vote in a doodle

The screenshot shows the Postman application interface. A PUT request is being made to `http://127.0.0.1:5000/doodles/1`. The request body is set to `JSON (application/json)` and contains the following JSON payload:

```
1 - {  
2 -   "person": "fred",  
3 -   "option": "1"  
4 - }
```

The response status is `200 OK`, time `8 ms`, and size `164 B`. The response body is:

```
1 - {  
2 -   "winners": [  
3 -     "1"  
4 -   ]  
5 - }
```

19)

The screenshot shows the Postman application interface. A PUT request is being made to `http://127.0.0.1:5000/doodles/1`. The request body is set to `JSON (application/json)` and contains the following JSON payload:

```
1 - {  
2 -   "person": "fred",  
3 -   "option": "2"  
4 - }
```

The response status is `200 OK`, time `14 ms`, and size `168 B`. The response body is:

```
1 - {  
2 -   "winners": [  
3 -     "2"  
4 -   ]  
5 - }
```

20)

The screenshot shows the Postman application interface. A PUT request is being made to `http://127.0.0.1:5000/doodles/1`. The request body is set to `JSON (application/json)` and contains the following JSON payload:

```
1: { "person": "barney",
2:   "option": "1"
3: }
```

The response status is `200 OK`, time `10 ms`, and size `164 B`. The response body is:

```
1: {
2:   "winners": [
3:     "1"
4:   ]
5: }
```

21)Get votes from a person that voted

The screenshot shows the Postman application interface. A GET request is being made to `http://127.0.0.1:5000/doodles/1/fred`. The request has an `Authorization` header set to `Inherit auth from parent`. A note states: `This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.` The response status is `200 OK`, time `7 ms`, and size `173 B`. The response body is:

```
1: {
2:   "votedoptions": [
3:     "1"
4:   ]
5: }
```

## 22) Get votes from a person that has not yet voted

The screenshot shows the Postman application interface. A collection named 'Doodles' is selected. A specific request titled '[6] - Retrieve preferences of a person in poll id' is displayed. The method is set to 'GET' and the URL is 'http://127.0.0.1:5000/doodles/1/wilma'. The 'Authorization' tab shows 'Inherit auth from parent'. The response body is shown in JSON format: { "votedoptions": [] }. The status bar at the bottom indicates a 200 OK response with 8 ms time and 166 B size.

## 23) Delete votes from a person that voted

The screenshot shows the Postman application interface. A collection named 'Doodles' is selected. A specific request titled '[7] - Delete preferences expressed by a person' is displayed. The method is set to 'DELETE' and the URL is 'http://127.0.0.1:5000/doodles/1/fred'. The 'Authorization' tab shows 'Inherit auth from parent'. The response body is shown in JSON format: { "removed": true }. The status bar at the bottom indicates a 200 OK response with 7 ms time and 163 B size.

## 24) Get votes from a person that was deleted

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for 'New', 'Import', 'Runner', and a search bar. Below the navigation is a toolbar with icons for 'Filter', 'History', 'Collections' (which is selected), 'Doodles' (with 7 requests), and others. The main workspace shows a collection named 'Doodles'. A specific request is selected: a GET request to 'http://127.0.0.1:5000/doodles/1/fred'. The 'Authorization' tab is active, showing 'Inherit auth from parent'. The response status is '200 OK' with a size of '166 B'. The response body is displayed as a JSON object: 

```
[{"votedoptions": []}]
```

## 25) Delete votes from a person that did not vote

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for 'New', 'Import', 'Runner', and a search bar. Below the navigation is a toolbar with icons for 'Filter', 'History', 'Collections' (which is selected), 'Doodles' (with 7 requests), and others. The main workspace shows a collection named 'Doodles'. A specific request is selected: a DELETE request to 'http://127.0.0.1:5000/doodles/1/wilma'. The 'Authorization' tab is active, showing 'Inherit auth from parent'. The response status is '200 OK' with a size of '164 B'. The response body is displayed as a JSON object: 

```
{"removed": false}
```

## 26) Retrieve votes from a non-existing poll

The screenshot shows the Postman application interface. In the top navigation bar, there are buttons for 'New', 'Import', 'Runner', and a search bar. The main workspace shows a collection named 'Doodles' containing 7 requests. A specific request is selected for viewing: a GET request to `http://127.0.0.1:5000/doodles/13/fred`. The request details pane shows the URL and method. The authorization tab indicates 'Inherit auth from parent'. The response pane displays a JSON object with 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.', and 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.'. The status bar at the bottom right shows 'Status: 404 NOT FOUND Time: 11 ms Size: 454 B'.

## 27) Get votes from a previously existing poll

This screenshot is identical to the previous one, showing a failed GET request to `http://127.0.0.1:5000/doodles/2/fred`. The response pane now displays a JSON object with code 410, description 'The requested URL is no longer available on this server and there is no forwarding address. If you followed a link from a foreign page, please contact the author of this page.', and message '410 Gone: The requested URL is no longer available on this server and there is no forwarding address. If you followed a link from a foreign page, please contact the author of this page.' The status bar at the bottom right shows 'Status: 410 GONE Time: 8 ms Size: 502 B'.

## [4 – Extra tests]

### 28) Invalid data structure provided as 'person'

The screenshot shows the Postman application interface. A PUT request is being made to `http://127.0.0.1:5000/doodles/1`. The request body is set to `JSON (application/json)` and contains the following invalid JSON:`1 - {  
2 | "person": "[\"test\"]",  
3 | "option": "1"  
4 }`The response status is `400 BAD REQUEST`, with a message indicating the browser sent a request that the server could not understand.

## 29) Invalid data structure provided while creating the doodle:

The screenshot shows the Postman application interface. A POST request is being made to `http://127.0.0.1:5000/doodles`. The request body is set to `JSON (application/json)` and contains the following invalid JSON:`1 - {  
2 | "title": "polls",  
3 | "options": "pizza"  
4 }`The response status is `400 BAD REQUEST`, with a message indicating the browser sent a request that the server could not understand.