

1 – APPLICATION SOURCE CODE, DOCUMENTATION AND DEPLOYMENT

The source code can be found at <https://github.com/leombro/ase-fall-17/tree/master/homework1>. Methods are adequately commented and a simple Javadoc is available in the /doc/ subfolder, both on GitHub and in the Moodle submission. Moreover, in the main folder of the submission there's a Postman collection (in JSON format) of tests for the various functionalities of the app (see section 3).

The application has been deployed to the Heroku service at the following link:

<https://leombro-ase-lab-17-hw1.herokuapp.com>.

2 – IMPLEMENTATION CHOICES

I decided to treat the two classes provided by the teachers (DOODLE.JAVA and VOTE.JAVA) as an “external library”, without changing their source code. Through testing on a “basic” version of my implementation and on the <https://toydoodle.herokuapp.com> app, I noticed some incorrect behavior; here's a list of them, together with my proposed fixes.

1. When performing PUT /doodles/ with a JSON body request that contains only the title field, the app throws a NullPointerException instead of ignoring the request gracefully.¹
Solution. Check that the Spring-generated Doodle object has both the title and options fields non-null.
2. The PUT /doodles/{id}/vote operation can modify an already-existing vote. (It shouldn't.)
Solution. Check that the name indicated in the JSON body request hasn't voted yet.
3. When performing PUT /doodles/{id}/vote with a JSON body request that doesn't contain the name field, the app inserts a vote anyway (with name = null).²
Solution. Check that the name field is present in the JSON body request.
4. The POST /doodles/{id}/vote/{name} operation can insert a new vote. (It shouldn't.)
Solution. Check that the name indicated in the JSON body request has already voted.
5. A “malicious” user can update another user's vote by means of an incorrect JSON body request when performing the POST /doodles/{id}/vote/{name} operation.
Example: user “john” modifies the vote of another user “jack” by sending the JSON request { 'name': 'jack', 'option': '...' } in a POST /doodles/{id}/vote/john operation.
Solution. Check that both the JSON's and Vote's name fields coincide.

Of course, other good programming practices (ex. testing for null before invoking an object's method) have been adopted.

¹ This behavior is due to the fact that the constructors for the Doodle class iterate on the list of options in order to build the list of votes. The Doodle(Doodle doodle) constructor copies the list of options from doodle without testing for null, so in the abovementioned scenario it tries to iterate on a null object (DOODLE.JAVA, line 28).

² The addVote(Vote vote) method doesn't check whether vote.getName() returns null (DOODLE.JAVA, line 49).

3 – TESTS

Here's a quick rundown of the app's functionalities. The screenshots include, where meaningful, the request and response bodies and, for all methods different from GET, the current status of the opened doodles.

1. Correct PUT /doodles

Creates a new Doodle.
(The JSON is well formatted and parses to a valid `Doodle` object.)

PUT `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/` Params

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▾

```
1 {
2   "title": "friday-meeting",
3   "options": [
4     "morning",
5     "lunchtime",
6     "afternoon"
7   ]
8 }
```

Body Cookies Headers (6) Test Results Status: 200

Pretty Raw Preview JSON ▾ ↺

```
1 0
```

```
1 {
2   "0": {
3     "title": "friday-meeting",
4     "options": [
5       "morning",
6       "lunchtime",
7       "afternoon"
8     ],
9     "votes": {
10      "afternoon": □,
11      "lunchtime": □,
12      "morning": □
13    }
14  }
15 }
```

2. Wrong PUT /doodles (no options field)

Tries to create a new Doodle without options, and fails.

PUT `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/` Params

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▾

```
1 {
2   "title": "sunday-meeting"
3 }
```

Body Cookies Headers (6) Test Results Status: 200

Pretty Raw Preview JSON ▾ ↺

```
1 -1
```

```
1 {
2   "0": {
3     "title": "friday-meeting",
4     "options": [
5       "morning",
6       "lunchtime",
7       "afternoon"
8     ],
9     "votes": {
10      "afternoon": □,
11      "lunchtime": □,
12      "morning": □
13    }
14  }
15 }
```

3. Correct PUT /doodles (with unnecessary fields)

Creates a new Doodle.
(The JSON is well formatted but it contains fields other than "title" and "options". It still parses to a valid Doodle object.)

PUT <https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/> Params

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "title": "saturday-dinner",
3   "options": [
4     "pizza",
5     "sushi",
6     "kebab"
7   ],
8   "transportation": [
9     "gino's car",
10    "bus",
11    "taxi"
12  ]
13 }
```

Body Cookies Headers (6) Test Results Status: 200

Pretty Raw Preview JSON

```
1 {
2   "0": {
3     "title": "friday-meeting",
4     "options": [
5       "morning",
6       "lunchtime",
7       "afternoon"
8     ],
9     "votes": {
10      "afternoon": ☐,
11      "lunchtime": ☐,
12      "morning": ☐
13    }
14  },
15   "1": {
16     "title": "saturday-dinner",
17     "options": [
18       "pizza",
19       "sushi",
20       "kebab"
21     ],
22     "votes": {
23       "pizza": ☐,
24       "sushi": ☐,
25       "kebab": ☐
26     }
27  }
28 }
```

4. GET /doodles

Retrieve all opened Doodles.

GET <https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/> Params

Pretty Raw Preview JSON

```
1 {
2   "0": {
3     "title": "friday-meeting",
4     "options": [
5       "morning",
6       "lunchtime",
7       "afternoon"
8     ],
9     "votes": {
10      "afternoon": ☐,
11      "lunchtime": ☐,
12      "morning": ☐
13    }
14  },
15   "1": {
16     "title": "saturday-dinner",
17     "options": [
18       "pizza",
19       "sushi",
20       "kebab"
21     ],
22     "votes": {
23       "pizza": ☐,
24       "sushi": ☐,
25       "kebab": ☐
26     }
27  }
28 }
```

5. GET /doodles/{id} (existing Doodle)

Retrieves the Doodle identified by `{id}`.
The Doodle is amongst those currently opened.

GET `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/0` Params

Body Cookies Headers (6) Test Results Status: 200

Pretty Raw Preview JSON

```
1 {
2   "title": "friday-meeting",
3   "options": [
4     "morning",
5     "lunchtime",
6     "afternoon"
7   ],
8   "votes": {
9     "afternoon": ☐,
10    "lunchtime": ☐,
11    "morning": ☐
12  }
13 }
```

6. GET /doodles/{id} (non-existing Doodle)

Retrieves the Doodle identified by `{id}`.
The Doodle is **NOT** amongst those currently opened.

GET `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/17` Params

Body Cookies Headers (5) Test Results Status: 200

Pretty Raw Preview Text

```
1
```

7. DELETE /doodles/{id}

Deletes the Doodle identified by `{id}`.
For simplicity, we only show the case where `{id}` refers to an existing Doodle.

DELETE `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/0` Params

Body Cookies Headers (5) Test Results Status: 200

Pretty Raw Preview Text

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

```
1 {
2   "1": {
3     "title": "saturday-dinner",
4     "options": [
5       "pizza",
6       "sushi",
7       "kebab"
8     ],
9     "votes": {
10      "pizza": ☐,
11      "sushi": ☐,
12      "kebab": ☐
13    }
14  }
15 }
```

8. PUT /doodles/{id}/vote (new vote)

Creates a new vote in the Doodle identified by `{id}`.
The JSON body request parses into a correct `Vote` object and `{id}` refers to an existing Doodle.

PUT `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/1/vote` Params

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json)

```
1 {
2   "name": "homer",
3   "option": "pizza"
4 }
```

Body Cookies Headers (6) Test Results Status: 200

Pretty Raw Preview Text

```
1 homer
```

```
1 {
2   "1": {
3     "title": "saturday-dinner",
4     "options": [
5       "pizza",
6       "sushi",
7       "kebab"
8     ],
9     "votes": {
10      "pizza": ☒,
11      "homer": ☐
12    },
13    "sushi": ☐,
14    "kebab": ☐
15  }
16 }
17
```

9. PUT /doodles/{id}/vote (invalid vote, "name" field missing)

Creates a new vote in the Doodle identified by `{id}`.
The JSON body request **does NOT** parse into a correct `Vote`; the `"name"` field is missing.

PUT `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/0/vote` Params

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json)

```
1 {
2   "option": "kebab"
3 }
```

Body Cookies Headers (5) Test Results Status: 200

Pretty Raw Preview Text

1

```
1 {
2   "1": {
3     "title": "saturday-dinner",
4     "options": [
5       "pizza",
6       "sushi",
7       "kebab"
8     ],
9     "votes": {
10      "pizza": [
11        "homer"
12      ],
13      "sushi": [],
14      "kebab": []
15    }
16  }
17 }
```

10. PUT /doodles/{id}/vote (correct new vote with unnecessary fields)

Creates a new vote in the Doodle identified by `{id}`.
The `{id}` refers to an existing Doodle. The JSON body request contains unnecessary fields, but still parses into a correct `Vote` object.

PUT `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/1/vote` Params

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json)

```
1 {
2   "name": "ralph",
3   "option": "sushi",
4   "failed-english": "impossible"
5 }
```

Body Cookies Headers (6) Test Results Status: 200 OK

Pretty Raw Preview Text

1 ralph

```
1 {
2   "1": {
3     "title": "saturday-dinner",
4     "options": [
5       "pizza",
6       "sushi",
7       "kebab"
8     ],
9     "votes": {
10      "pizza": [
11        "homer"
12      ],
13      "sushi": [
14        "ralph"
15      ],
16      "kebab": []
17    }
18  }
19 }
```

11. PUT /doodles/{id}/vote (user has already voted in that Doodle)

Creates a new vote in the Doodle identified by `{id}`.
The `{id}` refers to an existing Doodle. The JSON body request correctly parses into a correct `Vote` object, but the user has already voted once in this Doodle.

PUT `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/1/vote`

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json)

```
1 {
2   "name": "homer",
3   "option": "kebab"
4 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Text

1

```
1 {
2   "1": {
3     "title": "saturday-dinner",
4     "options": [
5       "pizza",
6       "sushi",
7       "kebab"
8     ],
9     "votes": {
10      "pizza": [
11        "homer"
12      ],
13      "sushi": [
14        "ralph"
15      ],
16      "kebab": []
17    }
18  }
19 }
```

12. GET /doodles/{id}/vote/{name} (existing vote)

Retrieves the vote casted by user `name` in the Doodle identified by `id`.
User `name` has already casted their vote in this Doodle.

GET `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/1/vote/ralph` Params

Body Cookies Headers (6) Test Results Status: 200

Pretty Raw Preview Text

```
1 sushi
```

13. GET /doodles/{id}/vote/{name} (non-existing vote)

Retrieves the vote casted by user `name` in the Doodle identified by `id`.
User `name` has NOT casted their vote in this Doodle.

GET `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/1/vote/quimby` Params

Body Cookies Headers (5) Test Results Status: 200

Pretty Raw Preview Text

```
1
```

14. DELETE /doodles/{id}/vote/{name}

Deletes the vote casted by user `name` in the Doodle identified by `id`.
For simplicity, `id` refers to an existing Doodle and `name` has already voted in it.

DELETE `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/1/vote/ralph` Params

Body Cookies Headers (5) Test Results Status: 200

Pretty Raw Preview Text

```
1
```

```
1 {
2   "1": {
3     "title": "saturday-dinner",
4     "options": [
5       "pizza",
6       "sushi",
7       "kebab"
8     ],
9     "votes": {
10      "pizza": [
11        "homer"
12      ],
13      "sushi": [],
14      "kebab": []
15    }
16  }
17 }
```

15. POST /doodles/{id}/vote/{name} (valid update)

Changes the vote casted by user `name` in the Doodle identified by `id`.
`id` refers to an existing Doodle and `name` has already voted in it; the `Vote` object refers to the same user as the one in the URI.

POST `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/1/vote/homer` Params

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json)

```
1 {
2   "name": "homer",
3   "option": "kebab"
4 }
```

Body Cookies Headers (6) Test Results Status: 200 OK

Pretty Raw Preview Text

```
1 homer
```

```
1 {
2   "1": {
3     "title": "saturday-dinner",
4     "options": [
5       "pizza",
6       "sushi",
7       "kebab"
8     ],
9     "votes": {
10      "pizza": [],
11      "sushi": [],
12      "kebab": [
13        "homer"
14      ]
15    }
16  }
17 }
```

16. POST /doodles/{id}/vote/{name} (wrong user)

Changes the vote casted by user `{name}` in the Doodle identified by `{id}`.
`{id}` refers to an existing Doodle and `{name}` has already voted in it, but the `{Vote}` object DOES NOT refer to the same user as the one in the URI.

POST `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/1/vote/homer` Params Send

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "name": "quimby",
3   "option": "kebab"
4 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 117

Pretty Raw Preview Text

1

```
1 {
2   "1": {
3     "title": "saturday-dinner",
4     "options": [
5       "pizza",
6       "sushi",
7       "kebab"
8     ],
9     "votes": {
10      "pizza": {},
11      "sushi": {},
12      "kebab": [
13        "homer"
14      ]
15    }
16  }
17 }
```

17. POST /doodles/{id}/vote/{name} (new vote)

Changes the vote casted by user `{name}` in the Doodle identified by `{id}`.
`{id}` refers to an existing Doodle, the `{Vote}` object refers to the same user as the one in the URI, but user `{name}` has NOT casted their vote yet.

POST `https://leombro-ase-lab-17-hw1.herokuapp.com/doodles/1/vote/quimby` Params Send

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "name": "quimby",
3   "option": "kebab"
4 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 1

Pretty Raw Preview Text

1

```
1 {
2   "1": {
3     "title": "saturday-dinner",
4     "options": [
5       "pizza",
6       "sushi",
7       "kebab"
8     ],
9     "votes": {
10      "pizza": {},
11      "sushi": {},
12      "kebab": [
13        "homer"
14      ]
15    }
16  }
17 }
```