

Campsite_Reservation_REST_API_Spring Boot-Junit-Mockito

Campsite_Reservation Web Services Application. This project is build in Spring boot 2 + Rest API + Junit and Mockito

Steps :

mvn clean

mvn test

mvn clean install

Go to the target folder

java -jar Campsite_Reservation_App-0.0.1-SNAPSHOT.jar

Verify your RESTful call. URL :

1. Get request for Available dates : localhost:8080/campsite/dates

Response :

```
{
  "responseCode": 200,
  "responseMessage": "Availabe Dates successfully fetched..!",
  "errorDiscription": null,
  "httpHeaders": null,
  "responseObject": "[Nov 13] [Nov 14] [Nov 15] [Nov 16] [Nov 17] [Nov 18] [Nov 19] [Nov 20] [Nov 21] [Nov 22] [Nov 23] [Nov 24] [Nov 25] [Nov 26] [Nov 27] [Nov 28] [Nov 29] [Nov 30] [Dec 01] [Dec 02] [Dec 03] [Dec 04] [Dec 05] [Dec 06] [Dec 07] [Dec 08] [Dec 09] [Dec 10] [Dec 11] [Dec 12] ",
  "error": false
}
```

}

2) Post request for Reservation : localhost:8080/campsite/reservation

Input Request :

```
{ "name": "Ashish Patel", "emailAddress": "patel.ashish610@gmail.com",  
  "startDate": "2018-11-13", "endDate": "2018-11-14" }
```

Response :

```
{ "responseCode": 200, "responseMessage": "The Campsite reservation successfully  
created..! The ReservationId is :-> 4", "errorDescription": null, "httpHeaders": {  
  "Location": [ "http://localhost:8080/campsite/reservation" ] }, "responseObject": null,  
  "error": false }
```

3) Update request for Modify Reservation : localhost:8080/campsite/reservation/3

Input Request :

```
{ "name": "ashish patel1234", "emailAddress": "ashish.patel@gmail.com",  
  "startDate": "2018-11-15", "endDate": "2018-11-16" }
```

Response :

```
{ "responseCode": 200, "responseMessage": "Your reservation has been modified  
successfully..! ", "errorDescription": null, "httpHeaders": { "Location": [  
  "http://localhost:8080/campsite/reservation/3" ] }, "responseObject": null, "error": false }
```

4) Delete request for Delete Reservation : localhost:8080/campsite/reservation/3

Screen Shot :

1) Get request for Available dates

The screenshot shows a REST client interface with a GET request to `localhost:8080/campsite/dates`. The request is sent, and the response is displayed in the Body tab. The response status is 200 OK, with a time of 115 ms and a size of 673 B. The response body is a JSON object with the following structure:

```
1 {
2   "responseCode": 200,
3   "responseMessage": "Availabe Dates fetched successfully!",
4   "errorDiscription": null,
5   "httpHeaders": null,
6   "responseObject": [
7     "[ Nov 14 ]",
8     "[ Nov 15 ]",
9     "[ Nov 16 ]",
10    "[ Nov 17 ]",
11    "[ Nov 18 ]",
12    "[ Nov 19 ]",
13    "[ Nov 20 ]",
14    "[ Nov 21 ]",
15    "[ Nov 22 ]",
16    "[ Nov 23 ]",
17  ]
18 }
```

2) Post request for create reservation :

localhost:8080 localhost:8080 localhost:8080 localhost:8080 localhost:8080 localhost:8080 localhost:8080 + ... No Environment

POST localhost:8080/campsite/reservation Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies

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

```
1 {
2   "name": "Ashish Patel",
3   "emailAddress": "patel.aswddwdwhish610@gmail.com",
4   "startDate": "2018-11-17",
5   "endDate": "2018-11-18"
6 }
```

Body Cookies Headers (3) Test Results Status: 201 Created Time: 11 ms Size: 3

Pretty Raw Preview JSON

```
1 {
2   "responseCode": 200,
3   "responseMessage": "The Campsite reservation successfully created..! The ReservationId is :-> 1",
4   "errorDiscription": null,
5   "httpHeaders": {
6     "Location": [
7       "http://localhost:8080/campsite/reservation"
8     ]
9   },
10  "responseObject": null,
11  "error": false
12 }
```

localhost:8080 localhost:8080 localhost:8080 localhost:8080 localhost:8080 localhost:8080 localhost:8080 + ... No Environment

POST localhost:8080/campsite/reservation Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

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

```
1 {
2   "name": "Ashish Patel",
3   "emailAddress": "patel.aswddwdwhish610@gmail.com",
4   "startDate": "2018-11-34",
5   "endDate": "2018-11-18"
6 }
```

Body Cookies Headers (4) Test Results Status: 500 Internal Server Error Time: 10 ms Size: 466 B

Pretty Raw Preview JSON

```
1 {
2   "responseCode": 422,
3   "responseMessage": "Campsite Reservation or Modification validation Fail.",
4   "errorDiscription": "Your StartDate or EndDate Format is Incorrect. Please follow - yyyy-mm-dd",
5   "httpHeaders": {
6     "Location": [
7       "http://localhost:8080/campsite/reservation"
8     ]
9   },
10  "responseObject": null,
11  "error": true
12 }
```

localhost:8 localhost:8 localhost:8 localhost:8 localhost:8 localhost:8 localhost:8 + ... No Environment

POST localhost:8080/campsite/reservation Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

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

```
1 {
2   "name": "Ashish Patel",
3   "emailAddress": "patel.aswddwdwhish610@gmail.com",
4   "startDate": "2018-11-14",
5   "endDate": "2018-11-30"
6 }
```

Body Cookies Headers (4) Test Results Status: 500 Internal Server Error Time: 20 ms Size: 509 B

Pretty Raw Preview JSON

```
1 {
2   "responseCode": 422,
3   "responseMessage": "Campsite Reservation or Modification validation Fail.",
4   "errorDiscription": "Your Start and End Date combination is Invalid. You can book campsite for maximum 3 days and upto 1 month in advance",
5   "httpHeaders": {
6     "Location": [
7       "http://localhost:8080/campsite/reservation"
8     ]
9   },
10  "responseObject": null,
11  "error": true
12 }
```

localhost:8 localhost:8 localhost:8 localhost:8 localhost:8 localhost:8 localhost:8 + ... No Environment

POST localhost:8080/campsite/reservation Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

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

```
1 {
2   "name": "Ashish Patel",
3   "emailAddress": "patel.aswddwdwhish610@gmail.com",
4   "startDate": "2018-11-17",
5   "endDate": "2018-11-18"
6 }
```

Body Cookies Headers (4) Test Results Status: 500 Internal Server Error Time: 9 ms Size: 500 B

Pretty Raw Preview JSON

```
1 {
2   "responseCode": 422,
3   "responseMessage": "Campsite Reservation or Modification validation Fail.",
4   "errorDiscription": "Sorry !! This Time frame is already reserved for another customer. Please try to reserve some other dates.!",
5   "httpHeaders": {
6     "Location": [
7       "http://localhost:8080/campsite/reservation"
8     ]
9   },
10  "responseObject": null,
11  "error": true
12 }
```

3) Put request for update reservation :

The screenshot shows a REST client interface with a PUT request to `localhost:8080/campsite/reservation/1`. The request body is a JSON object:

```
1 {
2   "name": "ashish patel1234",
3   "emailAddress": "ashish.patel@gmail.com",
4   "startDate": "2018-11-17",
5   "endDate": "2018-11-18"
6 }
```

The response status is **500 Internal Server Error** (Time: 23 ms, Size: 452 B). The response body is a JSON object:

```
1 {
2   "responseCode": 404,
3   "responseMessage": "Reservation Id not Found",
4   "errorDiscription": "Your ReservationId is not available in our System.! Please enter correct reservationId",
5   "httpHeaders": {
6     "Location": [
7       "http://localhost:8080/campsite/reservation/1"
8     ]
9   },
10  "responseObject": null,
11  "error": true
12 }
```

Below this, another screenshot shows a PUT request to `localhost:8080/campsite/reservation/0`. The request body is a JSON object:

```
1 {
2   "name": "ash12ish patel1234",
3   "emailAddress": "ashish.patel@gmail.com",
4   "startDate": "2018-11-19",
5   "endDate": "2018-11-20"
6 }
```

The response status is **200 OK** (Time: 16 ms, Size: 358 B). The response body is a JSON object:

```
1 {
2   "responseCode": 200,
3   "responseMessage": "Your reservation has been modified successfully..! ",
4   "errorDiscription": null,
5   "httpHeaders": {
6     "Location": [
7       "http://localhost:8080/campsite/reservation/0"
8     ]
9   },
10  "responseObject": null,
11  "error": false
12 }
```

4) Delete request for cancel reservation :

The screenshot shows the Postman interface for a DELETE request to `localhost:8080/campsite/reservation/3`. The request is configured with the method `DELETE` and the environment set to `No Environment`. The `Authorization` tab is selected, showing the type `Inherit auth from parent`. The `Body` tab is also selected, displaying the response body in JSON format.

Request Details:

- Method: `DELETE`
- URL: `localhost:8080/campsite/reservation/3`
- Environment: `No Environment`
- Authorization: `Inherit auth from parent`

Response Details:

- Status: `500 Internal Server Error`
- Time: `16 ms`
- Size: `452 B`

Response Body (JSON):

```
1 {
2   "responseCode": 404,
3   "responseMessage": "Reservation Id not Found",
4   "errorDiscription": "Your ReservationId is not available in our System.! Please enter correct reservationId",
5   "httpHeaders": {
6     "Location": [
7       "http://localhost:8080/campsite/reservation/3"
8     ]
9   },
10  "responseObject": null,
11  "error": true
12 }
```

The screenshot also shows a second instance of the Postman interface below, displaying a DELETE request to `localhost:8080/campsite/reservation/0`. The response status is `204 No Content`, with a time of `14 ms` and a size of `110 B`. The response body is empty.