This file contains the testcases for various scenarios of the project Fuber.

1. Test case - 1

Scenario : Customer wants to check the available cabs near him for a ride and has set on the pink preference as true.

Expected Output - The cab details of the nearest cab to the customer is identified and sent as a JSON. .

Sample URI : [http://localhost:8080/cabs-available/true?latitude=12.885105&longitude=77.645504](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)

Http method : GET

Parameters passed :

* preference for pink - set as true.
* Lat Long values of the customer.

Sample Output :

Details of the cab which is nearest (within 3 kms radius of customer)is given as response which is a pink cab. JSON object of the cab is received within response object.

{"cabID":110,"cabCurrLocation":{"latitude":12.88411,"longitude":77.6476},"userData":{"custID":null,"custCurrLocation":{"latitude":0.0,"longitude":0.0},"tripDetails":{"startPoint":{"latitude":0.0,"longitude":0.0},"endPoint":{"latitude":0.0,"longitude":0.0},"tripFare":0.0,"tripDistance":0.0,"startTime":null,"endTime":null,"totalTime":0.0}},"custLocation":null,"pinkColor":true,"carAvailability":true,"bookingStatus":false}

2. TestCase 2

Scenario : Customer wants to check the available cabs near him for a ride and has set on the pink preference as true.

Expected output : In the cars-available call, preferred cab doesnt come up. Throws an cabNotFoundException.

Sample URI : [http://localhost:8080/cabs-available/true?latitude=12.975105&longitude=77.245504](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)

Http method : GET

Parameters passed :

* preference for pink - set as true.
* Lat Long values of the customer.

Sample Output :

{

"message": "Sorry the requested cab is unavailble right now. Please check again",

"details": [

"Cab is not available"

]

}

3. TestCase 3

Scenario : Customer wants to check the available cabs near him for a ride.Pink perferance is not set here.

Expected Output : Customer gets a cab. In this case, the code checks for cabs irrespective of pink color. So there might be chances of pink cab getting allotted. JSON of Cabdata is returned.

Sample URI : <http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504>

Http method : GET

Parameters passed :

* preference for pink - as true or false. In this case false.
* Lat Long values of the customer.

Sample Output :

Details of the cab which is nearest (within 3 kms radius of customer)is given as response. Pink cab is the nearest cab available. Thus it is suggested by the Code.

{"cabID":110,"cabCurrLocation":{"latitude":12.88411,"longitude":77.6476},"userData":{"custID":null,"custCurrLocation":{"latitude":0.0,"longitude":0.0},"tripDetails":{"startPoint":{"latitude":0.0,"longitude":0.0},"endPoint":{"latitude":0.0,"longitude":0.0},"tripFare":0.0,"tripDistance":0.0,"startTime":null,"endTime":null,"totalTime":0.0}},"custLocation":null,"pinkColor":true,"carAvailability":true,"bookingStatus":false}

4. Testcase 4

Scenario : Customer wants to check the available cabs near him for a ride.Pink perferance is not set here.

Expected Output : No cabs are available within the radius of customer. Throws a cabNotFoundException as the response.

Sample URI : [http://localhost:8080/cabs-available/false?latitude=12.575105&longitude=77.245504](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)

Http method : GET

Parameters passed :

* preference for pink is false.
* Lat Long values of the customer.

Sample Output :

{

"message": "Sorry the requested cab is unavailble right now. Please check again",

"details": [

"Cab is not available"

]

}

5. Testcase 5

Scenario : incorrect URL pattern is send to the services.The parameters are wrong

Expected Output : Throws a bad request Exception.

Sample URI : [http://localhost:8080/cabs-available/sarath?latitude=12.575105&longitude=77.245504](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)

Http method : GET

Parameters passed :

* preference for pink is passed wrongly
* Lat Long values of the customer.

Sample Output :

{

"message": "Bad Request.Check URI Parameters",

"details": [

"Failed to convert value of type 'java.lang.String' to required type 'boolean'; nested exception is java.lang.IllegalArgumentException: Invalid boolean value [sarath]"

]

}

6. TestCase 6

Scenario : incorrect http method is used

Expected Output : Throws a Method not allowed Exception.

Sample URI : [http://localhost:8080/cabs-available/true?latitude=12.575105&longitude=77.245504](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)

Http method : POST

Parameters passed :

* preference for pink is passed as true
* Lat Long values of the customer.

Sample Output :

{

"timestamp": "2019-10-13T21:29:32.818+0000",

"status": 405,

"error": "Method Not Allowed",

"message": "Request method 'POST' not supported",

"path": "/cabs-available/sarath"

}

7. TestCase 7

Scenario : Customer wants to book the cab nearest.

Expected Output : Cab gets booked. The data passed as parameters are persisted to the model and in the response object, the distance of the cab from the location is displayed.

Sample URI : [http://localhost:8080/](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)book-cab

Http method : POST

Parameters passed :

* Cabid - 110 -> as request header
* Customer data - > as request body

{

"custID": 12345,

"custCurrLocation": {

"latitude": 12.884105,

"longitude": 77.645504

},

"tripDetails": {

"startPoint": {

"latitude": 12.884105,

"longitude": 77.645504

},

"endPoint": {

"latitude": 12.485105,

"longitude": 77.648504

},

"tripFare": 0,

"tripDistance": 0.0,

"startTime": null,

"endTime": null

}

}

Sample Output : Cab gets booked

RESPONSE :

Cab Id 110 has been booked and is enroute,0.23 kms away

8. TestCase 8

Scenario : wrong JSON is send to the services.

Expected Output : Bad request Exception is thrown.

Sample URI : [http://localhost:8080/](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)book-cab

Http method : POST

Parameters passed :

* Cabid - 110 -> as request header
* Customer data - > as request body

{{

"custID": 12345,

"custCurrLocation": {

"latitude": 12.884105,

"longitude": 77.645504

},

"tripDetails": {

"startPoint": {

"latitude": 12.884105,

"longitude": 77.645504

},

"endPoint": {

"latitude": 12.485105,

"longitude": 77.648504

},

"tripFare": 0,

"tripDistance": 0.0,

"startTime": null,

"endTime": null

}

},"as" :12}

Sample Output :

{

"timestamp": "2019-10-13T21:42:40.308+0000",

"status": 400,

"error": "Bad Request",

"message": "JSON parse error: Unexpected character ('{' (code 123)): was expecting double-quote to start field name; nested exception is com.fasterxml.jackson.core.JsonParseException: Unexpected character ('{' (code 123)): was expecting double-quote to start field name\n at [Source: (PushbackInputStream); line: 2, column: 2]",

"path": "/book-cab"

}

9. Testcase 9

Scenario : the driver starts the trip.

Expected Output : A response string which says trip has began. The start time is set with this service call. Current location of the cab is set as the location of the customer. This is to update the location of the cab in the map.

Sample URI : [http://localhost:8080/](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)start-trip

Http method : POST

Parameters passed

* Cab id - 110

Sample output :

Started the trip

10. Test case 10

Scenario : The parameter is not passed

Expected Output : A bad request exception is thrown

Sample URI : [http://localhost:8080/](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)start-trip

Http method : POST

Parameters passed

Sample Output

{

"timestamp": "2019-10-13T21:49:29.602+0000",

"status": 400,

"error": "Bad Request",

"message": "Missing request header 'cabid' for method parameter of type Integer",

"path": "/start-trip"

}

11. TestCase 11

Scenario : The trip ends and end-trip service is called.

Expected Output : Total fare, distance and time are calculated. The current cab location is updated to the destination. Bookingstatus and caravailability attributes are reset.

Sample URI : [http://localhost:8080/](http://localhost:8080/cabs-available/false?latitude=12.885105&longitude=77.645504)end-trip

Http method : POST

Parameters passed

* Cabid - 104

Sample Output :

JSON of the cab data is generated and passed onto response.

{

"cabID": 104,

"cabCurrLocation": {

"latitude": 12.485105,

"longitude": 77.648504

},

"userData": {

"custID": 12345,

"custCurrLocation": {

"latitude": 0.0,

"longitude": 0.0

},

"tripDetails": {

"startPoint": {

"latitude": 12.884105,

"longitude": 77.645504

},

"endPoint": {

"latitude": 12.485105,

"longitude": 77.648504

},

"tripFare": 86.49,

"tripDistance": 43.09,

"startTime": "2019-10-14T03:22:21.923317",

"endTime": "2019-10-14T03:22:40.442543",

"totalTime": 0.31

}

},

"custLocation": {

"latitude": 0.0,

"longitude": 0.0

},

"pinkColor": false,

"carAvailability": true,

"bookingStatus": false