

Teja Garidepally (12)
Assignment 6
Advanced Software Engg

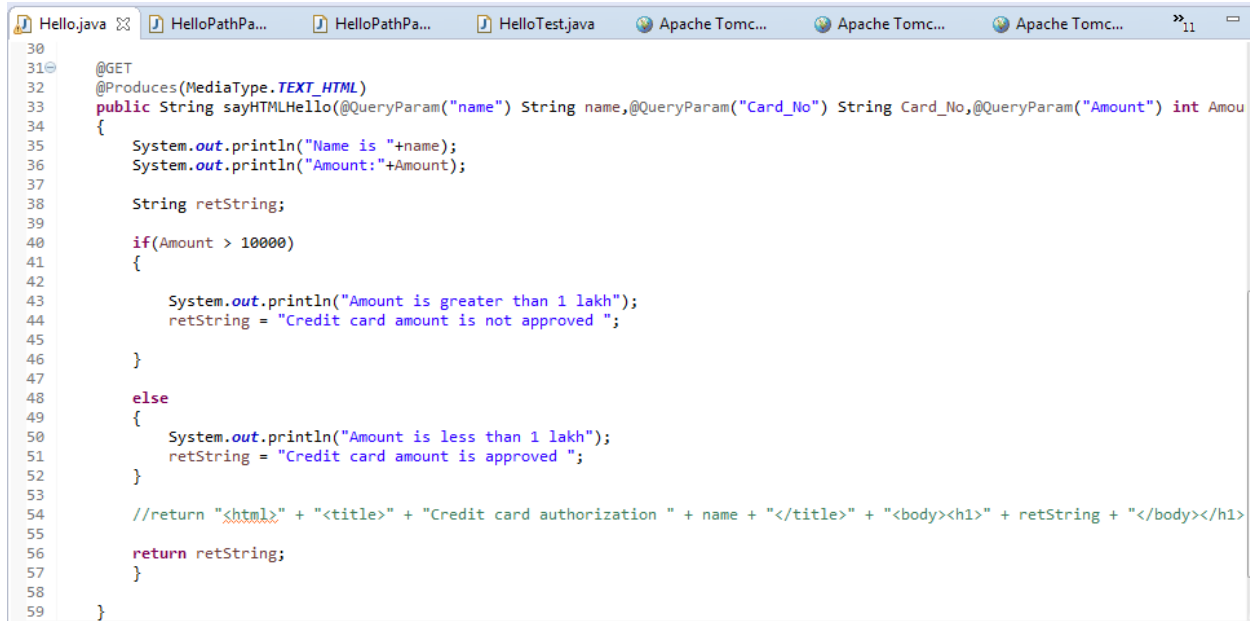
Description:

I have created two restful services using jersey REST API and written test cases for them.

Hello.java

This class takes input Emp name, card no and amount, It checks if the amount is less than 10000 or greater than 10000 and displays if the credit card is approved.

Credit amount is approved if amount is less than 1 Lakh and rejects if it is greater than 1 Lakh.

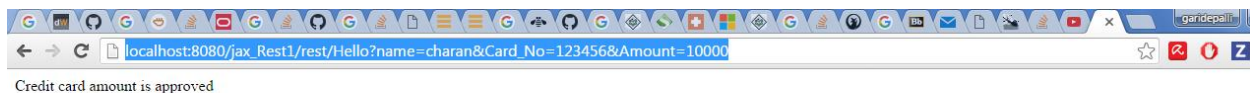


```
30
31 @GET
32 @Produces(MediaType.TEXT_HTML)
33 public String sayHTMLHello(@QueryParam("name") String name,@QueryParam("Card_No") String Card_No,@QueryParam("Amount") int Amount)
34 {
35     System.out.println("Name is "+name);
36     System.out.println("Amount:"+Amount);
37
38     String retString;
39
40     if(Amount > 10000)
41     {
42
43         System.out.println("Amount is greater than 1 lakh");
44         retString = "Credit card amount is not approved ";
45
46     }
47
48     else
49     {
50         System.out.println("Amount is less than 1 lakh");
51         retString = "Credit card amount is approved ";
52     }
53
54     //return "<html>" + "<title>" + "Credit card authorization " + name + "</title>" + "<body><h1>" + retString + "</body></h1>"
55
56     return retString;
57 }
58
59 }
```

By running this java file in server and opening the URL

http://localhost:8080/jax_Rest1/rest/Hello?name=charan&Card_No=123456&Amount=10000

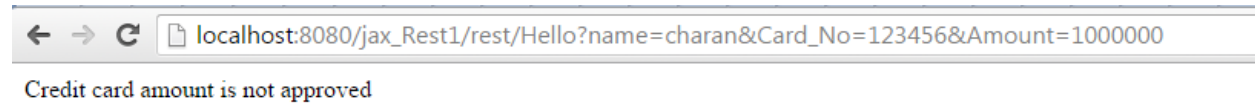
The above URL gives input to Hello.java



Credit card amount is approved

Gives result if credit card is approved or not based on amount given in URL

localhost:8080/jax_Rest1/rest/Hello?name=charan&Card_No=123456&Amount=1000000



HelloPathParam.java

This class takes input emp name,basic,hra,da,deductions and calculates total_pay and tax to be deducted



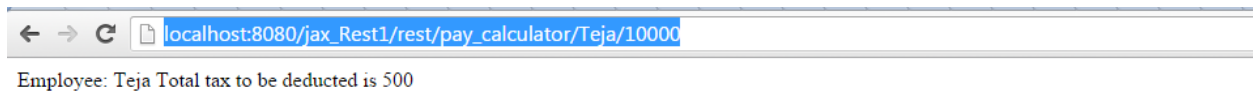
By running the below URL it gives total_pay for given input

http://localhost:8080/jax_Rest1/rest/pay_calculator/Teja/10000/1000/100/100



In the same way, running below URL gives tax to be deducted

http://localhost:8080/jax_Rest1/rest/pay_calculator/Teja/10000



JUnit Test cases:

HelloTest.java

It contains test cases for methods testSayPlaintextHello() and testSayHTMLHello() in

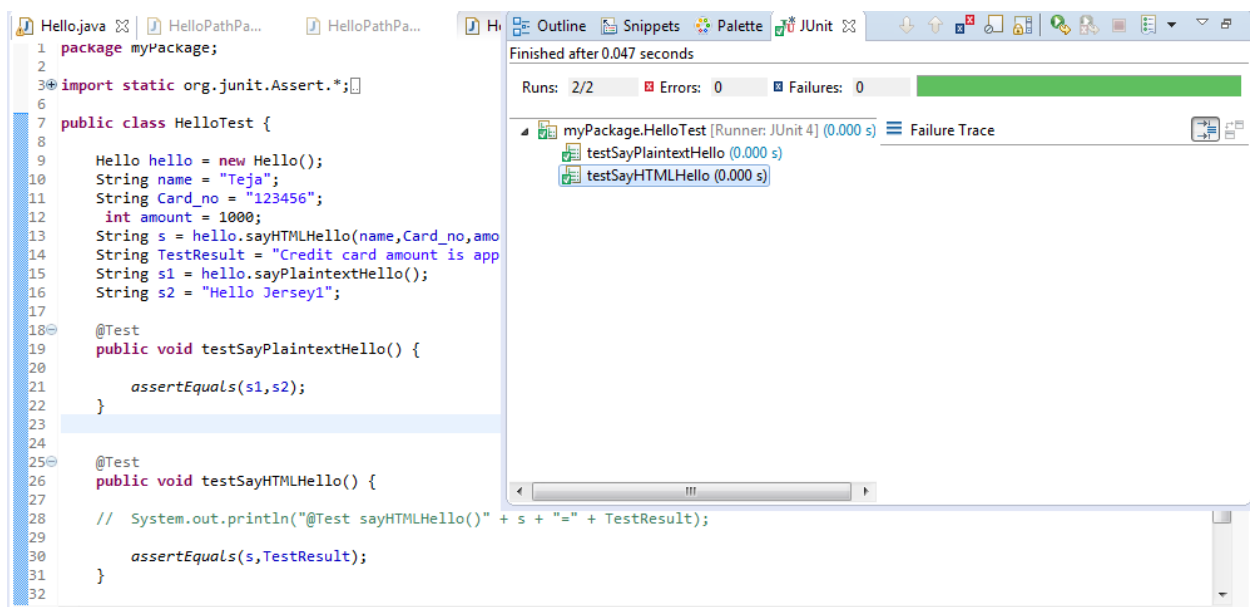
HelloTest.java



```
1 package myPackage;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class HelloTest {
8
9     Hello hello = new Hello();
10    String name = "Teja";
11    String Card_no = "123456";
12    int amount = 1000;
13    String s = hello.sayHTMLHello(name,Card_no,amount);
14    String TestResult = "Credit card amount is approved ";
15    String s1 = hello.sayPlaintextHello();
16    String s2 = "Hello Jersey1";
17
18    @Test
19    public void testSayPlaintextHello() {
20
21        assertEquals(s1,s2);
22    }
23
24
25    @Test
26    public void testSayHTMLHello() {
27
28        // System.out.println("@Test sayHTMLHello()" + s + "=" + TestResult);
29
30        assertEquals(s,TestResult);
31    }
32}
```

The two test methods created checks if the methods give correct output or not , by giving input to the methods and comparing the actual result with test result.

Success case:



```
1 package myPackage;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class HelloTest {
8
9     Hello hello = new Hello();
10    String name = "Teja";
11    String Card_no = "123456";
12    int amount = 1000;
13    String s = hello.sayHTMLHello(name,Card_no,amo
14    String TestResult = "Credit card amount is app
15    String s1 = hello.sayPlaintextHello();
16    String s2 = "Hello Jersey1";
17
18    @Test
19    public void testSayPlaintextHello() {
20
21        assertEquals(s1,s2);
22    }
23
24
25    @Test
26    public void testSayHTMLHello() {
27
28        // System.out.println("@Test sayHTMLHello()" + s + "=" + TestResult);
29
30        assertEquals(s,TestResult);
31    }
32}
```

Finished after 0.047 seconds

Runs: 2/2 Errors: 0 Failures: 0

myPackage.HelloTest [Runner: JUnit 4] (0.000 s)

- testSayPlaintextHello (0.000 s)
- testSayHTMLHello (0.000 s)

Failure cases:

The screenshot shows an IDE with a Java file named `HelloTest.java` and a JUnit runner window. The code defines a `Hello` class and a `HelloTest` class with two test methods. The first test, `testSayPlaintextHello`, passes. The second test, `testSayHTMLHello`, fails with a `org.junit.ComparisonFailure`. The console shows the output of the failing test.

```
1 package myPackage;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class HelloTest {
8
9     Hello hello = new Hello();
10    String name = "Teja";
11    String Card_no = "123456";
12    int amount = 10000000;
13    String s = hello.sayHTMLHello(name, Card_no, amo
14    String TestResult = "Credit card amount is app
15    String s1 = hello.sayPlaintextHello();
16    String s2 = "Hello Jersey1";
17
18    @Test
19    public void testSayPlaintextHello() {
20
21        assertEquals(s1, s2);
22    }
23
24
25    @Test
26    public void testSayHTMLHello() {
27
28        // System.out.println("@Test sayHTMLHello()" + s + "=" + TestResult);
29
30        assertEquals(s, TestResult);
31    }
32
```

JUnit Runner Summary:

- Finished after 0.063 seconds
- Runs: 2/2
- Errors: 0
- Failures: 1

Failure Trace:

- myPackage.HelloTest [Runner: JUnit 4] (0.000 s)
- testSayPlaintextHello (0.000 s)
- testSayHTMLHello (0.000 s)
- org.junit.ComparisonFailure: expected:<...edit card i...> but was:<Hello Jersey1>
- at myPackage.HelloTest.testSayHTMLHello(HelloTest.java:30)

Console Output:

```
<terminated> HelloTest [JUnit] C:\Program Files\Java\jdk1.8.0_66\bin\javaw.exe (Mar 2, 2016, 12:59:37 PM)
Amount:10000000
Amount is greater than 1 lakh
```

The screenshot shows an IDE with a Java file named `HelloTest.java` and a JUnit runner window. The code defines a `Hello` class and a `HelloTest` class with two test methods. The first test, `testSayPlaintextHello`, fails with a `org.junit.ComparisonFailure`. The second test, `testSayHTMLHello`, passes. The console shows the output of the failing test.

```
1 package myPackage;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class HelloTest {
8
9     Hello hello = new Hello();
10    String name = "Teja";
11    String Card_no = "123456";
12    int amount = 10000000;
13    String s = hello.sayHTMLHello(name, Card_no, amo
14    String TestResult = "Credit card amount is app
15    String s1 = hello.sayPlaintextHello();
16    String s2 = "Hello Jersey";
17
18    @Test
19    public void testSayPlaintextHello() {
20
21        assertEquals(s1, s2);
22    }
23
24
25    @Test
26    public void testSayHTMLHello() {
27
28        // System.out.println("@Test sayHTMLHello()" + s + "=" + TestResult);
29
30        assertEquals(s, TestResult);
31    }
32
```

JUnit Runner Summary:

- Finished after 0.047 seconds
- Runs: 2/2
- Errors: 0
- Failures: 2

Failure Trace:

- myPackage.HelloTest [Runner: JUnit 4] (0.000 s)
- testSayPlaintextHello (0.000 s)
- testSayHTMLHello (0.000 s)
- org.junit.ComparisonFailure: expected:<Hello Jersey> but was:<...edit card i...>
- at myPackage.HelloTest.testSayPlaintextHello(HelloTest.java:21)

Console Output:

```
<terminated> HelloTest [JUnit] C:\Program Files\Java\jdk1.8.0_66\bin\javaw.exe (Mar 2, 2016, 12:59:37 PM)
Amount:10000000
Amount is greater than 1 lakh
```

HelloPathParamTest.java:

This tests whether the responses given by HelloPathParam.java are correct or not based on given input.

