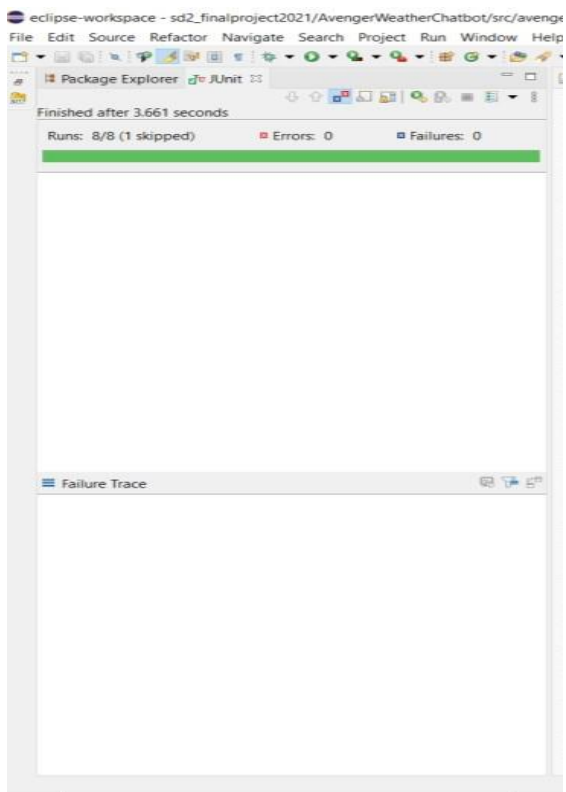


Result for TestWeather.java (Success)



Coding for TestWeather.java

```
1 package avenger.weather.weatherRequested;
2
3 import static org.junit.Assert.assertEquals;
4
12
13 public class TestWeather {
14
15     private Request testweather=new Request();
16
17     protected double temp;
18     protected String city;
19
20     public static void main(String[] args) {
21
22     }
23
24
25     //Test getDescription()
26     @Test
27     public void testdescription() {
28         Weather testdescription=testweather.weatherRequest();
29         assertEquals(testdescription.getDescription(),"broken clouds");
30     }
31
32     //Test getTemp()
33     @Test
34     public void testTemp() {
35         Weather testTemp=testweather.weatherRequest();
36         assertEquals("Error in testTemp",23.0,testTemp.getTemp(),0.1);
37     }
38
39     //Test Min()
40     @Test
41     public void testmin() {
42         Weather testmin=testweather.weatherRequest();
43         assertEquals("Error in testmin",23.0,testmin.getMin(),0.1);
44     }
45
46
47
48
49
50
51
52
53     //Test getfeels()
54     @Test
55     public void testfeels() {
56         Weather testfeels=testweather.weatherRequest();
57         assertEquals("Error in testfeels",21.38,testfeels.getFeelsLike(),0.1);
58     }
59
60     //Test getwindspeed()
61     @Test
62     public void testwindspeed() {
63         Weather testwindspeed=testweather.weatherRequest();
64         assertEquals("Error in getwindspeed",7.2,testwindspeed.getWindSpeed(),0.1);
65     }
66
67
68     //Test Location()
69     @Test
70     public void testlocation() {
71         Weather testlocation=testweather.weatherRequest();
72         assertEquals("error in testlocation", testlocation.getLocation(),testlocation.getLocation());
73     }
74 }
```

Test Runner at the end of Code

```
74
75 @Ignore
76 //Test Runner for Weather.class
77 @Test
78 public void main() {
79     Result result=JUnitCore.runClasses(Weather.class);
80     for (Failure failure : result.getFailures()) {
81         System.out.println(failure.toString());
82     }
83
84     System.out.println(result.wasSuccessful());
85 }
86
87
88
89 }
90
```

<

Writable

Smart Insert

1:1:0