

## Testing for JsonParser

```
1 package avenger.weather.jsonParser;
2
3
4@import static org.junit.Assert.*;
5
6 import org.junit.Before;
7 import org.junit.Ignore;
8 import org.junit.Test;
9 import org.junit.runners.Parameterized;
10 import org.junit.runner.RunWith;
11 import static org.junit.Assert.assertEquals;
12
13 import java.io.IOException;
14 import java.util.Map;
15
16 import avenger.weather.weatherRequested.Weather;
17
18 @RunWith(Parameterized.class)
19 public class Testjson {
20     Weather test;
21     Map<String, Object> map;
22
23     private jsonParser testjson=new jsonParser();
24     private Map<String, Object> respMap;
25     private Map<String, Object> mainMap;
26     private Map<String, Object> windMap;
27     private Map<String, Object> weatherMap;
28
29
30@    @Before
31        public void initialize() {
32            jsonParser testjs= new jsonParser();
33        }
34
35@    public Testjson(Map<String, Object> respMap, Map<String, Object> mainMap, Map<String, Object> windMap, Map<String, Object> weatherMap) {
36        this.respMap = respMap;
37        this.mainMap = mainMap;
38        this.windMap = windMap;
39        this.weatherMap = weatherMap;
40    }
41
42@    public Map<String, Object> getrespmap() {
43        return respMap;
44    }
45
46
47
48
49
50@    public Map<String, Object> getmainmap() {
51        return mainMap;
52    }
53
54@    public Map<String, Object> getwindmap() {
55        return windMap;
56    }
57
58
59@    public Map<String, Object> getweathermap() {
60        return weatherMap;
61    }
62
63
64
65@    public Map<String, Object> RespMap() {
66        Map<String, Object> respMap= Testjson.this.getrespmap();
67        return respMap;
68    }
69
70
71@    public Map<String, Object> MainMap() {
72        Map<String, Object> mainMap= Testjson.this.getmainmap();
73        return mainMap;
74    }
75
76
77@    public Map<String, Object> WindMap() {
78        Map<String, Object> windMap= Testjson.this.getwindmap();
79        return windMap;
80    }
81
82
83@    public Map<String, Object> WeatherMap() {
84        Map<String, Object> weatherMap= Testjson.this.getweathermap();
85        return weatherMap;
86    }
87
88@    @Test
89        public void testrespmap() throws IOException {
90            System.out.println(Testjson.this.RespMap());
91            assertEquals("error", Testjson.this.RespMap());
92        }
93    }
```

```
86     }
87
88     @Test
89     public void testmainmap() throws IOException {
90         System.out.println(Testjson.this.MainMap());
91         assertEquals("error", Testjson.this.MainMap());
92     }
93
94     @Test
95     public void testwindmap() throws IOException {
96         System.out.println(Testjson.this.WindMap());
97         assertEquals("error", Testjson.this.WindMap());
98     }
99
100    @Test
101    public void testweathermap() throws IOException{
102        System.out.println(Testjson.this.WeatherMap());
103        assertEquals("error", Testjson.this.WeatherMap());
104    }
105
106    @Ignore
107    @Test(expected=NullPointerException.class)
108    public void nullException() {
109        String a = null;
110        testjson.jsonStringParser(a);
111    }
112
113
114 }
115
```