

MappingServiceTest.java

```
1 package Test;
2
3 import static org.junit.Assert.*;
15
16 /* Mapping avoids Multiple occurrences of same literal values in multiple forms or
   multiple places
19 /* */
20
21
22 public class MappingServiceTest {
23     List<ConfigAttribute> configAttributes;
24
25
26     @Before
27     public void setUp() throws Exception {
28
29     }
30
31     /* This will validate the map method which map single value using the mappings
   */
32     @Test
33     public void mapTest() {
34         FormFactory.unID=0;
35         FormFactory formFactory=new FormFactory("configMapping.txt");
36         configAttributes=formFactory.csvParser();
37         String mappedValue;
38         mappedValue=MappingService.map(configAttributes, "1.0", 100);
39         assertEquals("ID", mappedValue);
40
41         mappedValue=MappingService.map(configAttributes, "1.1", 100);
42         assertEquals("How old are you?", mappedValue);
43
44         mappedValue=MappingService.map(configAttributes, "1.0", 200);
45         assertEquals("Id", mappedValue);
46
47         mappedValue=MappingService.map(configAttributes, "1.1", 200);
48         assertEquals("Age", mappedValue);
49     }
50
51     /* This will validate the mapArray method which map array values using the
   mappings */
52     @Test
53     public void mapArrayTest() {
54         FormFactory.unID=0;
55         FormFactory formFactory=new FormFactory("configMapping.txt");
56         configAttributes=formFactory.csvParser();
57         String[] labelNames={"1.0", "1.1"};
58
59         String[] mappedArray=MappingService.mapArray(configAttributes, labelNames,
   200, true);
60         assertEquals("Id", mappedArray[0]);
61         assertEquals("Age", mappedArray[1]);
62
63         mappedArray=MappingService.mapArray(configAttributes, labelNames, 200,
   false);
```

MappingServiceTest.java

```
64     assertEquals("1.0", mappedArray[0]);
65     assertEquals("1.1", mappedArray[1]);
66
67 }
68
69 }
70
```