

```
1 package failedTest.screenshots;
2
3 import java.io.File;
4 import org.apache.commons.io.FileUtils;
5 import org.openqa.selenium.By;
6 import org.openqa.selenium.OutputType;
7 import org.openqa.selenium.TakesScreenshot;
8 import org.openqa.selenium.WebDriver;
9 import org.openqa.selenium.firefox.FirefoxDriver;
10 import org.testng.Assert;
11 import org.testng.ITestResult;
12 import org.testng.annotations.AfterMethod;
13 import org.testng.annotations.AfterTest;
14 import org.testng.annotations.BeforeTest;
15 import org.testng.annotations.Test;
16
17 public class ScreenshotFailedCases {
18
19     static WebDriver driver;
20
21     @BeforeTest
22     public static void init() {
23         System.setProperty("webdriver.gecko.driver", "./Drivers/geckodriver.exe");
24
25         // Initiate Fire fox browser
26         driver = new FirefoxDriver();
27
28         // Maximize the browser
29         driver.manage().window().maximize();
30
31         // Pass application url
32         driver.get("https://duckduckgo.com/");
33         System.out.println("BeforeTest");
34     }
35
36     @Test
37     public void captureCorrectScreenMethod() throws Exception {
38         String Text = driver.findElement(By.xpath("//*[@id='logo_homepage_link']")).getText
39         ();
40         // Verify the text on the landing page
41         Assert.assertTrue(Text.contains("About DuckDuckGo"));
42     }
43
44     @Test
45     public void captureIncorrectScreenMethod() throws Exception {
46         // Fail test by using incorrect XPath to find the search box
47         driver.findElement(By.xpath("//*[@name='qe']")).sendKeys("agile");
48     }
49
50     @AfterTest
51     public static void exit() {
52         // Close the WebPage
53         driver.quit();
54     }
55
56     // AfterMethod annotation - This method executes after every test execution
57     @AfterMethod
58     public void screenShot(ITestResult result) {
59
60
```

```
61         // ITestResult.FAILURE is equals to result.getStatus() then it enter into
62         // if condition
63
64         if (ITestResult.FAILURE == result.getStatus()) {
65             try {
66
67                 // To create reference of TakesScreenshot
68                 TakesScreenshot screenshot = (TakesScreenshot) driver;
69
70                 // Call method to capture screenshot
71                 File src = screenshot.getScreenshotAs(OutputType.FILE);
72
73                 // Copy files to specific location result.getName() will
74                 // return name of test case so that screenshot name will be same as test
75                 case name
76                 FileUtils.copyFile(src, new File("./Screenshots/" + result.getName() +
77                     System.currentTimeMillis() + ".png"));
78                 System.out.println("Successfully captured a screenshot");
79             } catch (Exception e) {
80                 System.out.println("Exception while taking screenshot " + e.getMessage
81                 ());
82             }
83         }
84     }
85 }
```