Complete Rep

|  |
| --- |
|  |

|  |
| --- |
| **Highlight a particular element**  public static void highlightElement(WebElement element) { for (int i = 0; i <2; i++) { JavascriptExecutor js = (JavascriptExecutor) WebDriverManager.driver;    js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, "color: blue; border: 5px edger blue;");  js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, "");}} |

|  |
| --- |
| Objects.java  Storing locators in this file and calling them in the functions |

|  |
| --- |
| **Implicit Wait**  driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ; |

|  |
| --- |
| **Wait Until for two or more elements**  wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//\*[@id='FirstElement' or @id='SecondElement']"))); |

|  |
| --- |
| **Wait until if one of locator is located**  public static ExpectedCondition<WebElement> oneOfElementsLocatedVisible(By... args){ final List<By> byes = Arrays.asList(args); return new ExpectedCondition<WebElement>() { @Override public Boolean apply(WebDriver driver){ for (By by : byes){ WebElement el; try {el = driver.findElement(by);} catch (Exception r) {continue;} if (el.isDisplayed()) return el;} return false;}};}  Wait wait = new WebDriverWait(driver, Timeouts.WAIT\_FOR\_PAGE\_TO\_LOAD\_TIMEOUT);  WebElement webElement = (WebElement) wait.until( Helper.oneOfElementsLocatedVisible( By.xpath(SERVICE\_TITLE\_LOCATOR), By.xpath(ATTENTION\_REQUEST\_ALREADY\_PRESENTS\_WINDOW\_LOCATOR) )); |

|  |
| --- |
| **Inserting Timestamp to the screenshot file name to get the screenshot and reports accumulated :-**  import java.sql.Timestamp;  import java.text.SimpleDateFormat;  import java.util.Date;  public String returnCurrentTimestamp() {  private static final SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd-HH-mm-ss");  Timestamp timestamp = new Timestamp(System.currentTimeMillis());  return (sdf.format(timestamp));  }} |

|  |
| --- |
| import java.io.File;  import org.apache.commons.io.FileUtils;  import org.openqa.selenium.By;  import org.openqa.selenium.OutputType;  import org.openqa.selenium.TakesScreenshot;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.firefox.FirefoxDriver;  import org.testng.annotations.Test;    @Test  public void ElementPresent(By elementId) throws Exception {  driver.manage().timeouts().implicitlyWait(5,TimeUnit.SECONDS) ;    try{  **// Step 1 - Insert wait until check highlight check and implicit wait at start of each method**  wait.until(ExpectedConditions.presenceOfElementLocated(elementId));  **// Step 2 - Insert highlight for every element**  HighlightElement(elementId);      } catch (Exception e){  System.out.println("I'm in exception");  **// Step 3, 4 - Take screenshot**  getscreenshot();  System.printStackTrace();    } }  **// Step 1 - highlight element method**  public static void highlightElement(WebElement element) { for (int i = 0; i <2; i++) { JavascriptExecutor js = (JavascriptExecutor) WebDriverManager.driver;  js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, "color: yellow; border: 5px edger blue;");  js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, ""); } }  **// Step 2 -Get screenshot method**  public void getscreenshot() throws Exception {  File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);  FileUtils.copyFile(scrFile, new File("D:\\screenshot\_"+ returnCurrentTimestamp() +".png"));  } }  **// Step 3,4 - Get time stamp method**  public String returnCurrentTimestamp() {  private static final SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd-HH-mm-ss");Timestamp timestamp = new Timestamp(System.currentTimeMillis());  return (sdf.format(timestamp));  }} |

|  |
| --- |
| **Pre-requisites to Generate Extent Reports:**   1. Java should be installed (Link to Install and setup Java ) 2. TestNG should be installed 3. Extent Report Jars (Version 2.41.2) – [Download](https://drive.google.com/file/d/0ByJmgAhaLx0GVzktVFNNUEZPeWc/view) 4. extent-config.xml – It allows to configure HTML Report |

|  |
| --- |
| **Steps To Generate Extent Reports:**   1. Firstly, create a TestNG project in eclipse 2. Now download extent library files from the following link: [http://extentreports.relevantcodes.com/](https://drive.google.com/file/d/0ByJmgAhaLx0GVzktVFNNUEZPeWc/view) 3. Add the downloaded library files in your project 4. Create a java class say ‘ExtentReportsClass’ and add following code to it |

|  |
| --- |
| package softwareTestingMaterial;  import java.io.File;  import org.testng.Assert;  import org.testng.ITestResult;  import org.testng.SkipException;  import org.testng.annotations.AfterMethod;  import org.testng.annotations.AfterTest;  import org.testng.annotations.BeforeTest;  import org.testng.annotations.Test;    import com.relevantcodes.extentreports.ExtentReports;  import com.relevantcodes.extentreports.ExtentTest;  import com.relevantcodes.extentreports.LogStatus;    public class ExtentReportsClass{  ExtentReports extent;  ExtentTest logger;      @BeforeTest  public void startReport(){  extent = new ExtentReports (System.getProperty("user.dir") +"/test-output/STMExtentReport.html", true);  extent  .addSystemInfo("Host Name", "SoftwareTestingMaterial")  .addSystemInfo("Environment", "Automation Testing")  .addSystemInfo("User Name", "Sunil Kumar G");    extent.loadConfig(new File(System.getProperty("user.dir")+"\\extent-config.xml"));  }    @Test  public void automationTest1(){  logger = extent.startTest("Automation Test 1");  Assert.assertTrue(true);  //To generate the log when the test case is passed  logger.log(LogStatus.PASS, "Test Case Passed is Automation Test 1");  }    @Test  public void automationTest2(){  logger = extent.startTest("failTest");  Assert.assertTrue(false);  logger.log(LogStatus.PASS, "Test Case (failTest) Status is passed");  }    @Test  public void skipTest(){  logger = extent.startTest("skipTest");  throw new SkipException("Skipping - This is not ready for testing ");  }    @AfterMethod  public void getResult(ITestResult result){  if(result.getStatus() == ITestResult.FAILURE){  logger.log(LogStatus.FAIL, "Test Case Failed is "+result.getName());  logger.log(LogStatus.FAIL, "Test Case Failed is "+result.getThrowable());  }else if(result.getStatus() == ITestResult.SKIP){  logger.log(LogStatus.SKIP, "Test Case Skipped is "+result.getName());  }  // ending test  //endTest(logger) : It ends the current test and prepares to create HTML report  extent.endTest(logger);  }  @AfterTest  public void endReport(){  extent.flush();  extent.close();  }  } |