**Reading Excel file**

File src = **new** File(System.*getProperty*("user.dir") + "\\testData.xlsx");

FileInputStream fis = **new** FileInputStream(src);

XSSFWorkbook wb = **new** XSSFWorkbook(fis);

XSSFSheet sheet1 = wb.getSheetAt(0);

**int** rowCount = sheet1.getLastRowNum();

**int** colCount = sheet1.getRow(0).getLastCellNum();

System.***out***.println(rowCount);

System.***out***.println(colCount);

**for** (**int** i = 0; i <= rowCount; i++) { **for** (**int** j = 0; j < colCount; j++)

{

String data = sheet1.getRow(i).getCell(j).getStringCellValue();

}

**Slider**

WebElement we= driver.findElement(By.xpath(".//\*[@id='container']/div/div[2]/div[2]/div/div[1]/div/div/div[2]/section/div[3]/div[1]/div[2]/div"));

new Actions(driver).dragAndDropBy(we, -100, 0).build().perform();\*/

**Opening new tab**

driver.findElement(By.cssSelector("body")).sendKeys(Keys.CONTROL +"t");

ArrayList<String> tabs = new ArrayList<String> (driver.getWindowHandles());

driver.switchTo().window(tabs.get(1));

driver.get("http://www.testingexcellence.com");

--------------------------------------------------------------------------------------

Set<String> tab\_handles = driver.getWindowHandles();

**int** number\_of\_tabs = tab\_handles.size();

**int** new\_tab\_index = number\_of\_tabs-1; driver.switchTo().window(tab\_handles.toArray()[new\_tab\_index].toString());

**Scroll**

JavascriptExecutor jse = (JavascriptExecutor)*driver*;

jse.executeScript("window.scrollBy(0,550)", "");

**Screen shot**

File srcfile= ((TakesScreenshot)*driver*).getScreenshotAs(OutputType.***FILE***);

FileUtils.*copyFile*(srcfile, **new** File("E:\\screenshots\\abc.png"));

Hash Maps

HashMap<String, String> map = **new** HashMap<String, String>();

map.put("Ishan", "QA1");

map.put("Kuldip", "QA2");

map.put("Himanshu", "QA3");

map.put("Nalin", "QA4");

System.***out***.println(map);

System.***out***.println(map.get("Ishan"));

**for** (Entry<String, String> s : map.entrySet()) {

System.***out***.println(s.getKey() + " : " + s.getValue());

**Add cookie**

Cookie name = **new** Cookie("clickout", "allow");

driver.manage().addCookie(name);

**header Links Match**

String allLinks = driver.findElement(By.*xpath*(".//\*[@id='header-menu-container']/div[3]")).getText();

String[] a = allLinks.split("\n");

System.***out***.println("Total number of Header links are " + a.length);

Thread.*sleep*(2000);

// Expected Header Links

String[] exp = { "WOHNEN", "SCHLAFEN", "ESSEN", "BAD", "KINDER", "FLUR", "BÜRO", "GARTEN", "LAMPEN","HEIMTEXTILIEN", "DEKORATION", "BAUMARKT", "SALE", "|", "WOHNIDEEN", "MARKEN" };

**for** (**int** i = 0; i < a.length; i++) {

**if** (a[i].equalsIgnoreCase(exp[i])) {

System.***out***.println("Text matched: " + a[i]);

} **else**

System.***out***.println("text not matched");

**Color selection**

String WohnenColr = Wohnen.getCssValue("color");

String WohnenHexcolr = Color.*fromString*(WohnenColr).asHex();

**Writing Excel**

File src = **new** File(System.*getProperty*("user.dir") + "\\testData.xlsx");

FileInputStream fis = **new** FileInputStream(src);

XSSFWorkbook wb = **new** XSSFWorkbook(fis);

XSSFSheet sheet = wb.createSheet("TestCases");

**for** (**int** i = 0; i <= 25; i++) {

**for** (**int** j = 0; j < 1; j++) {

sheet.createRow(i).createCell(0).setCellValue("Pass");

sheet.getRow(i).createCell(1).setCellValue("fail");

FileOutputStream fos = **new** FileOutputStream(src);

wb.write(fos);

}

}

System.***out***.println("Sheet Created Successfully");

wb.close();

}

**static** WebDriver *driver*;

**static** String *key*;

**static** String *value*;

**public** **static** **void** main(String[] args) {

String url = "https://www.moebel.de/betatesting.php";

System.*setProperty*("webdriver.chrome.driver","E:\\Automationdata\\chromedriver.exe");

DesiredCapabilities capabilities = DesiredCapabilities.*chrome*();

ChromeOptions options = **new** ChromeOptions();

options.addExtensions(**new** File("E:\\Automationdata\\extension\_1\_3\_1.crx"));

capabilities.setCapability(ChromeOptions.***CAPABILITY***, options);

LoggingPreferences logPrefs = **new** LoggingPreferences();

logPrefs.enable(LogType.***BROWSER***, Level.***ALL***);

capabilities.setCapability(CapabilityType.***LOGGING\_PREFS***, logPrefs);

*driver* = **new** ChromeDriver(capabilities);

*driver*.get(url);

*driver*.manage().window().maximize();

WebElement drpdwn = *driver*.findElement(By.*xpath*("html/body/div[1]/select"));

Select sl = **new** Select(drpdwn);

sl.selectByValue("qa1c");

*OmniTracking*();

*driver*.quit();

}

**public** **static** **void** OmniTracking() {

String url = "https://www.moebel.de/magazin";

*driver*.get(url);

System.***out***.println("Title of Home Page is: " + *driver*.getTitle());

HashMap<String, String> map = **new** HashMap<String, String>();

LogEntries logEntries = *driver*.manage().logs().get(LogType.***BROWSER***);

**for** (LogEntry entry : logEntries) {

**if** (entry.getMessage().contains("chrome-extension") && entry.getMessage().contains(": "))

map.put(entry.getMessage().split("\"")[1].split(":")[0].trim(),

entry.getMessage().split("\"")[1].split(": ")[1].trim()).

}

// -------------------------------Print Value of Omniture Tracking-----------------------------------

\* try { print\_map(map); } catch (IOException | AWTException e) {

\* e.printStackTrace(); }

// ------------------------------Comparing Actual With ExpectedValue----------------------------------

*compare\_specificvalue*(map, "Page Name", "magazin0");

// --------------------------------Printing value in Console---------------------------------------

**private** **static** **void** print\_map(HashMap<String, String> map) **throws** IOException, AWTException {

**for** (Entry<String, String> s : map.entrySet()) {

*key* = s.getKey();

*value* = s.getValue();

System.***out***.println(s.getKey() + " : " + s.getValue());

}

}

// --------------------------------comparing some specific value---------------------------------------

**public** **static** **void** compare\_specificvalue(HashMap<String, String> map, String key, String value) **throws** IOException {

**if** (map.containsKey(key) && map.get(key) != **null**) {

System.***out***.println(map.get(key));

**if** (map.get(key).contains(value)) {

System.***out***.println("Yes it contains the value for specific key");

}**else**{

System.***out***.println("No, Value doesn't match with expected");

}

}

}