-check for unique element in html dom using name language

$$("[name='language']")

-find how many element in html dom using option tag

$$("option")

-check for xpath unique or not html

$x("//\*[@id='www-wikipedia-org']/div[6]/div[3]")

-isDisPlayed() method check the visibility, not the presence( the element is in the Dom but it can be hidden, not hidden), if element is hidden, isDisPlayed() will return false, but presence will reture true

public class TestIsElementPresent {

public static WebDriver driver;

public static Boolean isElementPresent(By by) {

try {

driver.findElement(by);

r

return true;

} catch (Throwable t) {

// TODO: handle exception

return false;

}

}

}

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

-find all the link in page, print all url

List<WebElement> link = driver.findElements(By.tagName("a"));

System.out.println("total number of link is: " + link.size());

for (WebElement webElement : link) {

System.out.println(webElement.getText() + "Url " + webElement.getAttribute("href"));

}

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

-XPATH syntax

//tagName[@attribute='value'] -- Relative XPath, Absolute XPath, Partial XPath

ex: //input[@id='identifierId']

console: $x("//input[@id='identifierId']")

-concatenate xpath to get unique location

//input[@id='identifierId'][@name='identifierId'] or //input[@id='identifierId'and @name='identifierId']

-handle xpath with dynamic , changing everytime to prevent hacking

//input[starts-with(@id,'iden')]

//input[contains(@id,'iden')]

-go to parent of element in dom

$x("//input[@id='identifierId']/..") or $x("//input[@id='identifierId']/parent::div")

-from parent to child

$x("//div[@class='aXBtI Wic03c']/div") then div[1] or [2]

-go to preceding sibling, that are 2 div

$x("//div[@class='aXBtI Wic03c']/div[3]/preceding-sibling::div[2]") or div[1]

-or go to following sibling

$x("//div[@class='aXBtI Wic03c']/div[3]/following-sibling::div[2]") or div[1]

Note: the last "/" is your element in the hierachy

-find element on the page

$x("//\*[contains(text(),'Signin')]") or $x("//\*[text()='Signin']")

-absolute xpath

$x("/html/body/div[1]/div[1]/div[2]/div/div[2]/div/div/div[2]/div/div[1]/div/form/span/section/div/div/div[1]/div/div[1]/div/div[1]/input")

-partial xpath: remove everthing until shown only 1 element

$x("//section/div/div/div[1]/div/div[1]/div/div[1]/input")

-select css selector (Note: Css selectot can't go up from child to parent like xpath)

$$("input[id='identifierId']")

-can be concatenate as well like xpath

$$("input[id='identifierId'][type='email']")

$$("input[id^='identifie']") ----start with

$$("input[id$='identifie']") ----end with

$$("input[id\*='identifie']") ----contain with

#identifierId - with Id using "#" $$("#identifierId")

- tagName = input

<input type="email" class="whsOnd zHQkBf" jsname="YPqjbf" autocomplete="username" spellcheck="false" tabindex="0" aria-label="Email or phone" name="identifier" value="" autocapitalize="none" id="identifierId" dir="ltr" data-initial-dir="ltr" data-initial-value="">

$$("input#identifierId")

-using with class "." maybe only using with "div" only

ex: $$("div.className")

NOte: after each space " " we need to delete it and add "."

Ex: rFrNMe N3Hzgf jjwyfe vHVGub zKHdkd sdJrJc Tyc9J

= .rFrNMe.N3Hzgf.jjwyfe.vHVGub.zKHdkd.sdJrJc.Tyc9J

-tagName+Id+class+ concatenate attribute = $$("input#identifierId.whsOnd.zHQkBf[type='email']")

-go to child using cssSelector:

$$("div.aCsJod.oJeWuf > div > div:first-child") nth-child(1) or (2); last-child

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

set up webdriver

WebDriverManager.chromedriver().setup();

ChromeDriver cd = new ChromeDriver(); or WebDriver cd (WebDriver is an interface, and ChromeDriver implement it

cd.get("https://www.google.com/");

WebDriverManager.firefoxdriver().setup();

FirefoxDriver fd = new FirefoxDriver();

fd.get("https://www.google.com/");

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

- hard wait:

Thread.sleep(2000);

- implicit wait:

WebDriverManager.chromedriver().setup();

WebDriver driver = new ChromeDriver();

driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);

- explicit wait:

WebDriverWait driverWait = new WebDriverWait(driver, 25);

driverWait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//a[contains(@href, 'Normandie')]")));

Notes: driver.findElement(By.xpath("//a[contains(@href, 'Normandie')]")).click();

implicit wait for all element appear then click it, explicit asap element appear then click it, if 2 element with the same name, explicit will click the first element, while implicit may click different element

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

mouse hover use Actions class

WebDriver driver = new ChromeDriver();

Actions a = new Actions(driver);

a.moveToElement(driver.findElement(By.xpath("//a[@class='a-level-1'][contains(text(),'Clothing & Shoes']"))).build().perform();

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

travel through multiple element in xpath

//div[@id='industryDisplay']/div/div/div/label

//div[@id='industryDisplay']/div/div/div[1]/label

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

get data from excel

package testcases;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Iterator;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.util.NumberToTextConverter;

import org.apache.poi.xssf.usermodel.XSSFCell;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import org.omg.CORBA.PUBLIC\_MEMBER;

public class inputFromExcel {

// public static void main(String[] args) throws IOException {

public ArrayList<String> getdata(String testCaseName) throws IOException {

// TODO Auto-generated method stub

ArrayList<String> a = new ArrayList<String>();

FileInputStream fisFileInputStream = new FileInputStream("/Users/harrisontranimac/Desktop/TestDataExcel.xlsx");

XSSFWorkbook workbook = new XSSFWorkbook(fisFileInputStream);

int sheets = workbook.getNumberOfSheets();

for (int i = 0; i < sheets; i++) {

if (workbook.getSheetName(i).equalsIgnoreCase("TestData")) {

XSSFSheet sheet = workbook.getSheetAt(i);

Iterator<Row> rowsIterator = sheet.iterator();

Row firstRow = rowsIterator.next();

Iterator<Cell> ce = firstRow.cellIterator();

int k = 0;

int column = 0;

while (ce.hasNext()) {

Cell value = ce.next();

if(value.getStringCellValue().equalsIgnoreCase("Testcase")) {

//grab desire column

column = k;

}

k++;

}

// System.out.println(column);

while(rowsIterator.hasNext()) {

Row r = rowsIterator.next();

if(r.getCell(column).getStringCellValue().equalsIgnoreCase(testCaseName)) {

Iterator<Cell> cv = r.cellIterator();

while(cv.hasNext()) {

// System.out.println(cv.next().getStringCellValue());

// a.add(cv.next().getStringCellValue());

Cell c = cv.next();

if( c.getCellType() == XSSFCell.CELL\_TYPE\_STRING) {

a.add(c.getStringCellValue());

} else

{

a.add(NumberToTextConverter.toText(c.getNumericCellValue()));

}

}

}

}

}

}

return a;

}

}

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

implements ITestListener to capture fail test: capture pics... send to email or save to folders, using 1 class to capture screen shot

public class TestListener extends TestBase implements ITestListener {

@Override

public void onTestFailure(ITestResult result) {

// TODO Auto-generated method stub

String methodnameString = result.getName().trim();

try {

testUtil.captureScreenshot(methodnameString);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

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

screenshot path for Mac

mailscreenshotpath = System.getProperty("user.dir")+"/screenshot/"+methodName+"\_"+year+"\_"+date+"\_"+(month+1)+"\_"+day+"\_"+min+"\_" +sec+".jpeg";

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

set up remote webdriver selenium grid

public class TestParametersXML {

public WebDriver driver;

public DesiredCapabilities cap = new DesiredCapabilities();

@Parameters("browser")

@Test

public void launchBrowser( String browser) throws MalformedURLException {

Date date = new Date();

System.out.println(browser + " " + date.toString());

if (browser.equals("chrome")) {

cap.setPlatform(Platform.ANY);

cap.setBrowserName("chrome");

ChromeOptions options = new ChromeOptions();

options.merge(cap);

} else if (browser.equals("firefox")) {

cap.setPlatform(Platform.ANY);

cap.setBrowserName("firefox");

FirefoxOptions options = new FirefoxOptions();

options.merge(cap);

}

driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), cap);

driver.get("http://gmail.com");

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.findElement(By.id("identifierId")).sendKeys("harrison");

driver.quit();

}

}

-----------------------XML file -----------------------

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Parallel Suite" parallel="tests">

<test name="Firefox Test">

<parameter name="browser" value="firefox"></parameter>

<classes>

<class name="testcases.TestParametersXML"></class>

</classes>

</test> <!-- Test -->

<test name="Chrome Test">

<parameter name="browser" value="chrome"></parameter>

<classes>

<class name="testcases.TestParametersXML"></class>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

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

data driven, page object model, hybrid are approach

get the current file path:System.getProperty("user.dir")

current root: "./"

string replace old character with new character" d.toString().replace(":", "\_").replace(" ", "\_")

generate log with time stamp: .../app\_${current.date}\_log.txt

A screenshot of a computer

Description automatically generated

public static WebDriver *driver*;

public static Properties *config* = new Properties();

public static Properties *OR* = new Properties();

public static FileInputStream *fis*;

public static Logger *log* = org.apache.log4j.Logger.*getLogger*("devpinoyLogger");

*fis* = new FileInputStream(System.*getProperty*("user.dir") + "/src/test/resources/properties/Config.properties");

*config*.load(*fis*);

*log*.debug("config file loaded");

use css in web console: $$(“button[ng-click=’addCust()’]”)

Asse