Note: All the jars are added through maven dependency

1. Excel Reader Class:

package Utilities;

import org.apache.commons.compress.archivers.dump.InvalidFormatException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.\*;

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

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

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

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

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

public class FileHandlers {

// @Test

public static void writeExcel(String path, String FileName, String Sheet, String rowkey,String Header,String value) throws org.apache.poi.openxml4j.exceptions.InvalidFormatException, FileNotFoundException {

Workbook workbook;

FileInputStream file = new FileInputStream(new File(path + "\\" + FileName));

//Map<String, String> data = new HashMap<String, String>();

try {

if(FileName.contains(".xlsx"))

{

workbook= new XSSFWorkbook(file);

}else {

workbook = WorkbookFactory.create(new File(path + "\\" + FileName));

}

DataFormatter dataFormatter = new DataFormatter();

Sheet sheet = workbook.getSheet(Sheet);

for (Row row : sheet) {

if (!(row.getRowNum() == 0)) {

if (dataFormatter.formatCellValue(row.getCell(0)).equalsIgnoreCase(rowkey))

for (int i = 0; i < row.getLastCellNum(); i++) {

if(sheet.getRow(0).getCell(i).getStringCellValue().equals(Header)){

row.getCell(i).setCellValue(value);

}

}

}

}

file.close();

FileOutputStream outFile =new FileOutputStream(new File(path + "\\" + FileName));

workbook.write(outFile);

outFile.close();

workbook.close();

//return data;

} catch (Exception e) {

e.printStackTrace();

//return data;

}

}

public static Map<String, String> readExcel(String path, String FileName, String Sheet, String rowkey) throws org.apache.poi.openxml4j.exceptions.InvalidFormatException {

Workbook workbook;

Map<String, String> data = new HashMap<String, String>();

try {

if(FileName.contains(".xlsx"))

{

workbook= new XSSFWorkbook(new File(path + "\\" + FileName));

}else {

workbook = WorkbookFactory.create(new File(path + "\\" + FileName));

}

DataFormatter dataFormatter = new DataFormatter();

Sheet sheet = workbook.getSheet(Sheet);

for (Row row : sheet) {

if (!(row.getRowNum() == 0)) {

if (dataFormatter.formatCellValue(row.getCell(0)).equalsIgnoreCase(rowkey))

for (int i = 0; i < row.getLastCellNum(); i++) {

data.put(dataFormatter.formatCellValue(sheet.getRow(0).getCell(i)),

dataFormatter.formatCellValue(row.getCell(i)));

}

}

}

workbook.close();

return data;

} catch (Exception e) {

e.printStackTrace();

return data;

}

}

public static List<Map<String, String>> readExcelAllRow(String path, String FileName, String Sheet, String rowkey) throws org.apache.poi.openxml4j.exceptions.InvalidFormatException {

Workbook workbook;

Map<String, String> data;

List<Map<String, String>> lst = new ArrayList<Map<String, String>>();

try {

if(FileName.contains(".xlsx"))

{

workbook= new XSSFWorkbook(new File(path + "\\" + FileName));

}else {

workbook = WorkbookFactory.create(new File(path + "\\" + FileName));

}

DataFormatter dataFormatter = new DataFormatter();

Sheet sheet = workbook.getSheet(Sheet);

for (Row row : sheet) {

if (!(row.getRowNum() == 0)) {

data = new HashMap<String, String>();

for (int i = 0; i < row.getLastCellNum(); i++) {

data.put(dataFormatter.formatCellValue(sheet.getRow(0).getCell(i)),

dataFormatter.formatCellValue(row.getCell(i)));

}

lst.add(data);

}

}

workbook.close();

return lst;

} catch (Exception e) {

e.printStackTrace();

return lst;

}

}

}

1. ConfigFileReader class:

package Utilities;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStream;

import java.util.Properties;

public class ConfigProvider {

public static String configFileReader(String sKey) {

String sConfigFilePath = System.getProperty("user.dir") + "/src/selenium.properties";

Properties prop = new Properties();

InputStream input = null;

String sValue = null;

try {

input = new FileInputStream(sConfigFilePath);

prop.load(input);

sValue = prop.getProperty(sKey);

} catch (IOException ex) {

ex.printStackTrace();

} finally {

if (input != null) {

try {

input.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

return sValue.trim();

}

}

1. Properties file as selenium.properties:
2. ApplicationUrl=https://www.icicibank.com/calculator/fd-calculator.html
3. tmobile.home.url=https://www.t-mobile.com/
4. DefaultBrowser=Chrome
5. ChromeDriverPath=D:/Apps/Drivers/chromedriver.exe

4. Base Page Object or Driver class where we will initiate driver for different browsers:

package Utilities;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.NoSuchElementException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.openqa.selenium.support.ui.ExpectedCondition;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.FluentWait;

import org.openqa.selenium.support.ui.WebDriverWait;

import io.github.bonigarcia.wdm.DriverManagerType;

import io.github.bonigarcia.wdm.WebDriverManager;

public class BasePageObject {

protected WebDriver driver;

public BasePageObject() {

WebDriverManager.getInstance(DriverManagerType.CHROME).setup();

driver = new ChromeDriver();

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

driver.manage().timeouts().implicitlyWait(200, TimeUnit.SECONDS);

//driver.manage().timeouts().pageLoadTimeout(500, TimeUnit.SECONDS);

}

public BasePageObject(String BrowserName) {

if (BrowserName.equalsIgnoreCase("Chrome")) {

WebDriverManager.getInstance(DriverManagerType.CHROME).setup();

driver = new ChromeDriver();

}

else if(BrowserName.equalsIgnoreCase("Firefox")) {

WebDriverManager.getInstance(DriverManagerType.CHROME).setup();

driver = new FirefoxDriver();

}else {

WebDriverManager.getInstance(DriverManagerType.CHROME).setup();

driver = new InternetExplorerDriver();

}

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

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

//driver.manage().timeouts().pageLoadTimeout(500, TimeUnit.SECONDS);

}

public void navigateToUrl(String Url) {

driver.navigate().to(Url);

}

public void quit() {

driver.quit();

}

public WebElement getElementSafely(By path) {

FluentWait<WebDriver> wait = new FluentWait<WebDriver>(driver)

.withTimeout(240, TimeUnit.SECONDS)

.pollingEvery(5, TimeUnit.SECONDS)

.ignoring(NoSuchElementException.class);

WebElement el = wait.until(ExpectedConditions.visibilityOfElementLocated(path));

return el;

}

public void clickElementSafely(By path)

{

FluentWait<WebDriver> ClickableWait = new FluentWait<WebDriver>(driver)

.withTimeout(240, TimeUnit.SECONDS)

.pollingEvery(5, TimeUnit.SECONDS)

.ignoring(NoSuchElementException.class);

WebElement el = ClickableWait.until(ExpectedConditions.elementToBeClickable(driver.findElement(path)));

el.click();

}

public void clickElementSafely(By path,long timeoutInSeconds)

{

FluentWait<WebDriver> ClickableWait = new FluentWait<WebDriver>(driver)

.withTimeout(timeoutInSeconds, TimeUnit.SECONDS)

.pollingEvery(5, TimeUnit.SECONDS)

.ignoring(NoSuchElementException.class);

WebElement el = ClickableWait.until(ExpectedConditions.elementToBeClickable(driver.findElement(path)));

el.click();

}

}

5. Page Object for Homepage :

package tMobile.PO;

import org.openqa.selenium.By;

import Utilities.BasePageObject;

public class tMobileHomePo extends BasePageObject{

public tMobileHomePo(){

super();

}

public tMobileHomePo(String BrowserName){

super(BrowserName);

}

public enum FIELDS{

PhoneLink(By.xpath("//a[contains(text(),'Phones')]")),

FilterList(By.xpath("//button[@id='dropdownMenu1']")),

NewConditionCheckBox(By.xpath("(//span[contains(text(),'New')])[1]")),

AccessoriesLink(By.xpath("//span[contains(text(),'Accessories')]")),

priceFilter(By.xpath("")),

PriceHighToLowfilter(By.xpath("")),

FirstproductName(By.xpath("(//browse-tile)[1]/div/a")),

TMobileLogo(By.xpath("(//browse-tile)[1]/div/a"));

private By findby;

FIELDS(By findby){

this.findby = findby;

}

}

public void navigateToPhone() {

//driver.findElement(FIELDS.PhoneLink.findby);

clickElementSafely(FIELDS.PhoneLink.findby);

}

public void Filter(String type) {

clickElementSafely(FIELDS.FilterList.findby);

if(type.contains("New")) {

clickElementSafely(FIELDS.NewConditionCheckBox.findby);

}

}

public void clickOnAccessories() {

clickElementSafely(FIELDS.AccessoriesLink.findby);

}

}

6. Test Class

package Demo.PageObject;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.ui.Select;

import Utilities.ConfigProvider;

import io.github.bonigarcia.wdm.DriverManagerType;

import io.github.bonigarcia.wdm.WebDriverManager;

public class LoginPagePO {

private static WebDriver driver;

public LoginPagePO() {

WebDriverManager.getInstance(DriverManagerType.CHROME).setup();

// System.setProperty("webdriver.chrome.driver",ConfigProvider.configFileReader("ChromeDriverPath"));

driver = new ChromeDriver();

}

public LoginPagePO(String Browser) {

if(Browser.equals("Chrome"))

{

WebDriverManager.getInstance(DriverManagerType.CHROME).setup();

// System.setProperty("webdriver.chrome.driver",ConfigProvider.configFileReader("ChromeDriverPath"));

driver = new ChromeDriver();

}

else if(Browser.equals("Firefox")) {

WebDriverManager.getInstance(DriverManagerType.FIREFOX).setup();

driver = new FirefoxDriver();

}else {

WebDriverManager.getInstance(DriverManagerType.IEXPLORER).setup();

driver = new InternetExplorerDriver();

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

}

public WebDriver getDriver() {

return driver;

}

public enum FIELDS {

HelpLink(By.xpath("")), FD\_Type(By.id("ddlTypeOfFixedDeposit")), Amount(By.id("loanAmount")), lblMaturityValue(

By.id("spnMaturityValue")), LblAggregateInterest(By.id("spnAIAmount")),

DaysOnlyRadio(By.id("idDays")),

DaysInput(By.id("tenureDay")),companyLogo(By.xpath("//\*[@class='Logo']"));

private By findby;

FIELDS(By findby) {

this.findby = findby;

}

}

public void navigateToHomeUrl(String Url) throws InterruptedException {

driver.get(Url);

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

Thread.sleep(3000);

driver.switchTo().frame(driver.findElement(By.id("paymframe")));

}

public void clickOnCheckEmi() {

driver.findElement(FIELDS.HelpLink.findby).click();

}

public void selectfdType(String option) {

driver.findElement(FIELDS.FD\_Type.findby).sendKeys(option);

}

public void enterloanAmount(String amt) {

// System.out.println(driver.findElement(FIELDS.Amount.findby).getText());

driver.findElement(FIELDS.Amount.findby).click();

driver.findElement(FIELDS.Amount.findby).clear();

driver.findElement(FIELDS.Amount.findby).sendKeys(amt);

driver.findElement(FIELDS.Amount.findby).sendKeys(Keys.TAB);

}

public void getMaturityValue() {

System.out.println(driver.findElement(FIELDS.lblMaturityValue.findby).getText());

}

public void getAggregateInterest() {

System.out.println(driver.findElement(FIELDS.LblAggregateInterest.findby).getText());

}

public void selectDaysOnlyTenure() {

driver.findElement(FIELDS.DaysOnlyRadio.findby).click();

}

public void enterDays(String days) {

driver.findElement(FIELDS.DaysInput.findby).clear();

driver.findElement(FIELDS.DaysInput.findby).sendKeys(days);

driver.findElement(FIELDS.Amount.findby).sendKeys(Keys.TAB);

}

public void clickLogo()

{

driver.navigate().refresh();

driver.findElement(FIELDS.companyLogo.findby).click();

}

}