233. Download and uploading file using file attribute sendKeys with Selenium

So you need to come to this page and click on download.

if you update apple value to 350 in excel sheet and save it,

so you have updated the content. Now, go and upload this file. So what it does is it'll take the, again, latest content from this Excel and apply to this table in the UI.

// download – Edit Excel – Upload – wait for success message to show up and wait for disappear - verify updated excel data showing in the webtable.

when you are sending the keys, send the path where your file is sitting

in your local system, So my file is sitting in that download folder

Create one new class(uploadDownload)

WebDriverManager.*chromedriver*().setup();

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

driver.manage().window().fullscreen();

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

driver.get("https://rahulshettyacademy.com/upload-download-test/index.html");

//click on download button

driver.findElement(By.*cssSelector*("#downloadButton")).click();

// uploaded the file in the local system give the file place path

WebElement upload = driver.findElement(By.*cssSelector*("input[type='file']"));

upload.sendKeys("C:\\Users\\Hari.Sankar\\Downloads\\download.xlsx");

234. Handle Synchronized toast messages with explicit wait for appear & disappear

//wait for success message to show up and wait for disappear

By toastLocator = By.*cssSelector*(".Toastify\_\_toast-body div:nth-child(2");

WebDriverWait wait = **new** WebDriverWait(driver,10);

wait.until(ExpectedConditions.*visibilityOfElementLocated*(toastLocator))

String toastText = driver.findElement(toastLocator).getText();

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

Assert.*assertEquals*("Updated Excel Data Successfully.",toastText);

wait.until(ExpectedConditions.*invisibilityOfElementLocated*(toastLocator));

A screenshot of a computer

Description automatically generated

235. Write Smart xpath to find table row column cell based on the conditions

//verify updated excel data showing in the web table ( modify code manually )

String fruitName = "Apple";

String priceColumn= driver.findElement(By.*xpath*("//div[text()='Price']")).getAttribute("data-column-id");

String actualPrice = driver.findElement(By.*xpath*("//div[text()='"+fruitName+"']/parent::div/parent::div/div[@id='cell-"+priceColumn+"-undefined']")).getText();

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

Assert.*assertEquals*("350", actualPrice);

237. End to end solution for updating excel and uploading the file with validations

238 . code download

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.util.concurrent.TimeUnit;

**import** io.github.bonigarcia.wdm.WebDriverManager;

**import** java.io.IOException;

**import** java.time.Duration;

**import** java.util.ArrayList;

**import** java.util.Iterator;

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

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

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

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

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

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

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

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

**import** org.testng.Assert;

**public** **class** UploadDownload {

**public** **static** **void** main(String[] args) **throws** IOException { // **TODO** Auto-generated method stub

String fruitName = "Apple";

String updatedValue = "603";

String fileName = "C:\\Users\\Hari.Sankar\\Downloads\\download.xlsx";

WebDriverManager.*chromedriver*().setup();

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

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

driver.get("https://rahulshettyacademy.com/upload-download-test/index.html");

// download

driver.findElement(By.*cssSelector*("#downloadButton")).click();

// Edit excel - getColumnNumber of Price -getRownumber of Apple-> update excel // with row,col

**int** col = *getColumnNumber*(fileName, "price");

**int** row = *getRowNumber*(fileName, "Apple");

Assert.*assertTrue*(*updateCell*(fileName, row, col, updatedValue));

// upload

WebElement upload = driver.findElement(By.*cssSelector*("input[type='file']"));

upload.sendKeys(fileName);

// wait for success message to show up and wait for disappear

By toastLocator = By.*cssSelector*(".Toastify\_\_toast-body div:nth-child(2");

WebDriverWait wait = **new** WebDriverWait(driver,10);

wait.until(ExpectedConditions.*visibilityOfElementLocated*(toastLocator));

String toastText = driver.findElement(toastLocator).getText();

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

Assert.*assertEquals*("Updated Excel Data Successfully.", toastText);

wait.until(ExpectedConditions.*invisibilityOfElementLocated*(toastLocator));

// verify updated excel data showing in the web table

String priceColumn = driver.findElement(By.*xpath*("//div[text()='Price']")).getAttribute("data-column-id");

String actualPrice = driver.findElement(By.*xpath*("//div[text()='" + fruitName

+ "']/parent::div/parent::div/div[@id='cell-" + priceColumn + "-undefined']")).getText();

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

Assert.*assertEquals*(updatedValue, actualPrice);

}

**private** **static** **boolean** updateCell(String fileName, **int** row, **int** col, String updatedValue) **throws** IOException {

// **TODO** Auto-generated method

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

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

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

XSSFSheet sheet = workbook.getSheet("Sheet1");

Row rowField = sheet.getRow(row - 1);

Cell cellField = rowField.getCell(col - 1);

cellField.setCellValue(updatedValue);

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

workbook.write(fos);

workbook.close();

fis.close();

**return** **true**;

}

**private** **static** **int** getRowNumber(String fileName, String text) **throws** IOException {

// **TODO** Auto-generated method stub

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

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

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

XSSFSheet sheet = workbook.getSheet("Sheet1"); // Identify Testcases coloum by scanning the entire 1st row

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

// sheet is collection of rows

**int** k = 1;

**int** rowIndex = -1;

**while** (rows.hasNext()) {

Row row = rows.next();

Iterator<Cell> cells = row.cellIterator();

**while** (cells.hasNext()) {

Cell cell = cells.next();

**if** (cell.getCellType() == CellType.***STRING*** && cell.getStringCellValue().equalsIgnoreCase(text)) {

rowIndex = k;

}

}

k++;

}

**return** rowIndex;

}

**private** **static** **int** getColumnNumber(String fileName, String colName) **throws** IOException {

// **TODO** Auto-generated method stub

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

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

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

XSSFSheet sheet = workbook.getSheet("Sheet1");

// Identify Testcases coloum by scanning the entire 1st row

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

// sheet is collection of rows

Row firstrow= rows.next();

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

// row is collection of cells

**int** k = 1;

**int** coloumn = 0;

**while** (ce.hasNext()) {

Cell value = ce.next();

**if** (value.getStringCellValue().equalsIgnoreCase(colName)) {

coloumn = k;

}

k++;

}

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

**return** coloumn;

}

}