



Selenium Easy

For Complete Web Automation Testing Tutorials using Selenium Webdriver

Home >> Selenium Tutorials >> How to verify entries in Exported CSV file

How to verify entries in Exported CSV file

In this article we will look into verify Export functionality. Export functionality is exporting a document/downloading a document which has number of records / data which need to be verified. Generally exported document will be in the form of csv file. We will now read csv file with the help of Java FileReader and LineNumberReader

Export button will be placed to export the records/data to our local system from the application. If you observe the below image, you will see different options to export the data which is available in table.

The screenshot shows a web application interface with a table of employee data. At the top right, there are buttons for 'Copy', 'CSV', 'Excel', 'PDF', and 'Print'. The 'CSV' button is circled in blue. A red callout box points to it with the text 'Click on CSV to export file as CSV'. On the left, there is a 'Show 10 entries' dropdown menu. The table has columns: Name, Position, Office, Age, Start date, and Salary. The first 10 rows of data are visible. At the bottom left, a blue circle highlights the text 'Showing 1 to 10 of 57 entries'. A red callout box points to it with the text 'Need to get the total number of entries from here'. At the bottom right, there is a pagination bar with 'Previous', '1', '2', '3', '4', '5', '6', and 'Next'.

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008/11/28	\$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09	\$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12	\$86,000
Bradley Greer	Software Engineer	London	41	2012/10/13	\$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07	\$206,850
Brielle Williamson	Integration Specialist	New York	61	2012/12/02	\$372,000
Bruno Nash	Software Engineer	London	38	2011/05/03	\$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011/12/12	\$106,450
Cara Stevens	Sales Assistant	New York	46	2011/12/06	\$145,600
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	2012/03/29	\$433,060

We need to follow the below steps to verify the export functionality.

Step 1:- First get the number of entries that the table has.

Step 2:- Click on Export button (CSV in the case of below example).

Step 3:- Download the document to a specific folder or default downloads folder.

Step 4:- Read the csv file and get number of entries in Csv (Need to exclude the header when comparing entries)

Step 5:- Compare the entries in the table and number of entries in the csv file.

Now we will see the below example:-

```
package com.pack;

import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.LineNumberReader;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxProfile;
import org.testng.Assert;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

public class ExportExample {

    WebDriver driver;

    private static String downloadPath = "D:\\seleniumdownloads";
    private String URL="appURL";

    @BeforeClass
    public void testSetup() throws Exception{
        driver = new FirefoxDriver(firefoxProfile());
        driver.manage().window().maximize();
    }

    public static FirefoxProfile firefoxProfile() throws Exception {
        FirefoxProfile firefoxProfile = new FirefoxProfile();
        firefoxProfile.setPreference("browser.download.folderList",2);
        firefoxProfile.setPreference("browser.download.manager.showWhenStarting",false);
        firefoxProfile.setPreference("browser.download.dir",downloadPath);
        firefoxProfile.setPreference("browser.helperApps.neverAsk.saveToDisk",
"text/csv,application/x-msexcel,application/excel,application/x-excel,application/vnd.ms-excel,image/png,image/jpeg,text/html,text/plain,application/msword,application/xml");

        return firefoxProfile;
    }

    @Test
    public void testExportAllRecords() throws InterruptedException {
        driver.get(URL);
    }
}
```

```

        int noOfEntries = getNumberOfEntries();
        System.out.println("Total number of entries are :- "+noOfEntries);

        WebElement elementCSV = driver.findElement(By.xpath(".*//*[@id='ToolTables_example_1']/span[contains(text(),'CSV')]"));
        elementCSV.click();

        File file = getLatestFilefromDir(downloadPath);
        String csvFileName = file.getName();
        System.out.println("CSV File Downloaded is :- "+csvFileName);

        System.out.println("Verifying number of entries with number of entries in csv");
        Assert.assertEquals(noOfEntries, getRecordsCountInCSV(downloadPath,csvFileName));
    }

```

Below method is used to get the number of entries in the table as shown in the above image.

```

    public int getNumberOfEntries() {

        String entriesTxt = driver.findElement(By.id("example_info")).getText().trim();
        String[] aEntriesText = entriesTxt.split(" ");
        String totalEntriesText = aEntriesText[aEntriesText.length-2];
        return Integer.parseInt(totalEntriesText);
    }

```

Below is the method to read the CSV file and get the number of entries present in the exported csv file. It takes the file name as the parameter

```

        public int getRecordsCountInCSV(String downloadPath, String csvFileName)
        {
            int lineNumberCount = 0;
            try {
                if (!csvFileName.isEmpty() || csvFileName != null) {
                    String filePath = downloadPath + System.get
Property("file.separator") + csvFileName;
                    System.out.println(filePath);
                    File file = new File(filePath);
                    if (file.exists()) {
                        System.out.println("File found : " + csvFil
eName);

                        FileReader fr = new FileReader(file);
                        LineNumberReader lineNumberreader = new L
ineNumberReader(fr);

                        while (lineNumberreader.readLine() != nul
l) {

                            lineNumberCount++;
                        }
                        //To remove the header
                        lineNumberCount=lineNumberCount-1;
                        System.out.println("Total number of lines
found in csv : " + (lineNumberCount));
                        lineNumberreader.close();
                    } else {
                        System.out.println("File does not exist
");
                    }
                }
            }
        }
    }
}

```

Below method is used to get the latest file from the directory. It takes the folder path as the parameter and returns the file which is recently added to the folder.

```

private File getLatestFilefromDir(String dirPath){
    File dir = new File(dirPath);
    File[] files = dir.listFiles();
    if (files == null || files.length == 0) {
        return null;
    }

    File lastModifiedFile = files[0];
    for (int i = 1; i < files.length; i++) {
        if (lastModifiedFile.lastModified() < files[i].lastModified()) {
            lastModifiedFile = files[i];
        }
    }
    return lastModifiedFile;
}
}

```

Hope the article helped you. Please feel free to comment if need more information.

Selenium Tutorials:

Selenium Tutorials

Selenium Tutorials

- › Introduction to selenium
- › Basic HTML concepts required for Selenium
- › Locators for Selenium
- › How to Run Webdriver in chrome browser?
- › How to run webdriver in IE browser?
- › Selenium webdriver using Eclipse
- › Synchronization in Selenium Webdriver
- › XPath tutorial for Selenium
- › CSS selectors for Selenium with example
- › Firefox Not Connected Exception in Selenium
- › Handle windows popups using Selenium Webdriver
- › Working with Frames in Selenium Webdriver
- › How to handle javascript alerts, confirmation and prompts?
- › Defining Firefox Profile preferences using Selenium Webdriver
- › Why Automation Testing is required?
- › Importance of Automation Testing
- › Mouse Hover Actions in Selenium Webdriver
- › StaleElementReferenceExceptions in Selenium Webdriver
- › Difference between Webdriver get() and Webdriver navigate()
- › Navigation Methods in Webdriver with Examples
- › Handling Cookies in Webdriver
- › How to addcookie with Selenium Webdriver
- › How to delete Cookies in Selenium Webdriver
- › WebDriver Methods
- › Automation Test Frameworks
- › Taking Screenshot using Webdriver
- › Webdriver SELECT Methods to work with Dropdowns
- › Webdriver Select with Multiple Attribute
- › Handling keyboard events and mouse hover events using Webdriver
- › Set browser width and height in Selenium Webdriver

- › Selenium Automation Framework Example
- › UI Map or Object Repository using Properties File
- › Test Data in automation framework
- › Reporting Structure in Selenium Automation Framework
- › Test Configuration File / Executable File
- › Uploading a file with Selenium Webdriver
- › Upload file using AutoIT
- › Uploading a file with sendKeys method
- › Read data from Properties file using Java Selenium
- › Scrolling web page with Selenium Webdriver using java
- › Keyword Driven Framework Example
- › Install AutoIT in Windows
- › Examples with xpath and Css (ID, Name, Text and Links)
- › Testing in multiple browsers using selenium and testng
- › Working with Checkbox using Resuable Methods
- › Working with Select examples
- › File upload using Robots
- › Download file using selenium webdriver
- › Perform operations on new window
- › Introduction to Page Object Model Framework
- › Mouse hover using javaScript Executor
- › Working with Multiple Checkboxes with Safecheck
- › Click element using JavaScriptExecutor
- › Drag and Drop using Webdriver Action Class
- › WebDriver Waits Examples
- › Simple Page Object Model example
- › Double Click on element using Webdriver
- › Right Click Context Menu example
- › Working with AutoComplete Text box
- › Working with Date Picker Calendar
- › Working with Ajax controls using Webdriver
- › Get Attribute values using Webdriver
- › Get Css values using Webdriver
- › Resizing a web element using movebyoffset
- › Validate Downloaded file after clicking on downloaded button/ link
- › How to verify entries in Exported CSV file
- › Element is not clickable at point
SeleniumWebdriverException
- › Extract PDF text And Verify Text Present in PDF

using WebDriver

- › Handling Authentication Window with WebDriver (In Firefox, Chrome and IE)

Recent Post

Handling Authentication Window with WebDriver
(In Firefox, Chrome and IE)

Selenium Webdriver Tutorials

Locators for Selenium

Optional annotation in TestNG

Parameterization in TestNG using testng.xml

Automation Framework

- › Framework Structure
- › Object Repository
- › Test Data
- › Test Scripts
- › Reports
- › Configuration and Execution

Tags Cloud

Asserts (1) AutoComplete (1) Autolt (2) Automation (3) Build.xml (1) Checkboxes (2) Cookies (3) Css (2) Css Examples (2) DataProvider (1) Date Picker (1) Drag and Drop (1) Exceptions (4) File Download (3) Firefox Profile (2) Framework Frameworks (5) iFrames (1) Interview Questions (1) JavascriptExecutor (3) JUnit (2) Keyword Driven (1) Listeners (3) Manual Testing (1) Maven (5) Mouse Hover (2) Page Object Model (2) Page Scroll (1) Parallel Test (3) properties file (1)

Home

- › Home
- › Tutorials
- › About
- › Contact
- › Privacy Policy
- › Site Map

Tutorial Menu

- › Selenium Tutorials
- › TestNG Tutorial
- › JUnit Tutorial
- › JXL Tutorial

- » [Apache POI](#)
- » [ANT Tutorial](#)
- » [Maven Tutorial](#)

Selenium Tutorials

- » [Basic Webdriver Tutorials](#)
- » [Advanced Webdriver Tutorials](#)
- » [Selenium Framework Examples](#)
- » [JavascriptExecutor Examples](#)

Interview Questions

- » [Testing Concepts](#)
- » [Quality Standards](#)
- » [General Questions](#)
- » [Roles & Responsibilities](#)
- » [Technology & Environment](#)

Copyright © 2015 - All Rights Reserved : Selenium Easy

We welcome your questions and requests