

File Download POP UP

What is file download pop up?

- A file download pop up is a pop-up window that asks the user to download a program to their computer's hard drive.

Characteristics of file download pop up

- File download pop up characteristics is different with different browsers.
- In all the browsers, following are the common behavior,
 - We cannot inspect the file download pop up.

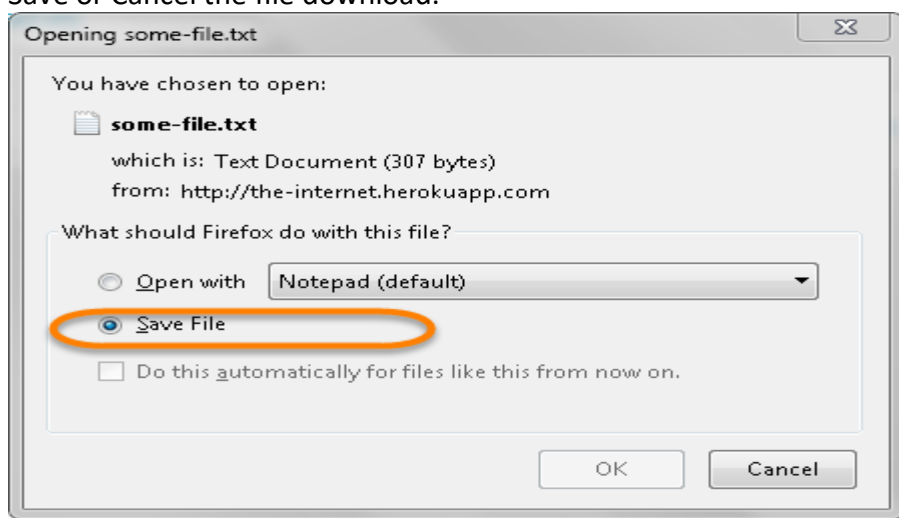
Note: Once click on the link of the file, one dialog box or confirmation window will appears and ask to save the file. If you try to inspect the element of the window, you will notice that it is not the browser HTML window but a system window, which cannot be handled by the Selenium. As selenium is just a Web Browser Automation Tool.

Different browsers can have different behaviours.

- **Chrome Behavior:** It will start downloading the file by default, as soon as the user click on the link of any file.
- **IE Behavior:** IE displays a bar at the bottom of the window and displays the option to Save or Cancel the file download.



- **Firefox Behavior:** This will display a dialog box window and displays the option to Save or Cancel the file download.



In order to handle file download pop up, we have 3 different ways. Following are the 3 different ways to handle the file download pop up.

1. Using AutoIT
2. Using Robot class
3. Using Firefox options class

Handling In **Firefox Browser**:

1. Using AutoIT
 - In order to perform any operation on the download pop up. We need to identify the elements.
 - But in this case, AutoIT window info tool will fail to identify the elements.
 - So, to handle this. We use the keyboard buttons to select the buttons like "TAB" multiple times to select "OK" or "CANCEL" buttons.

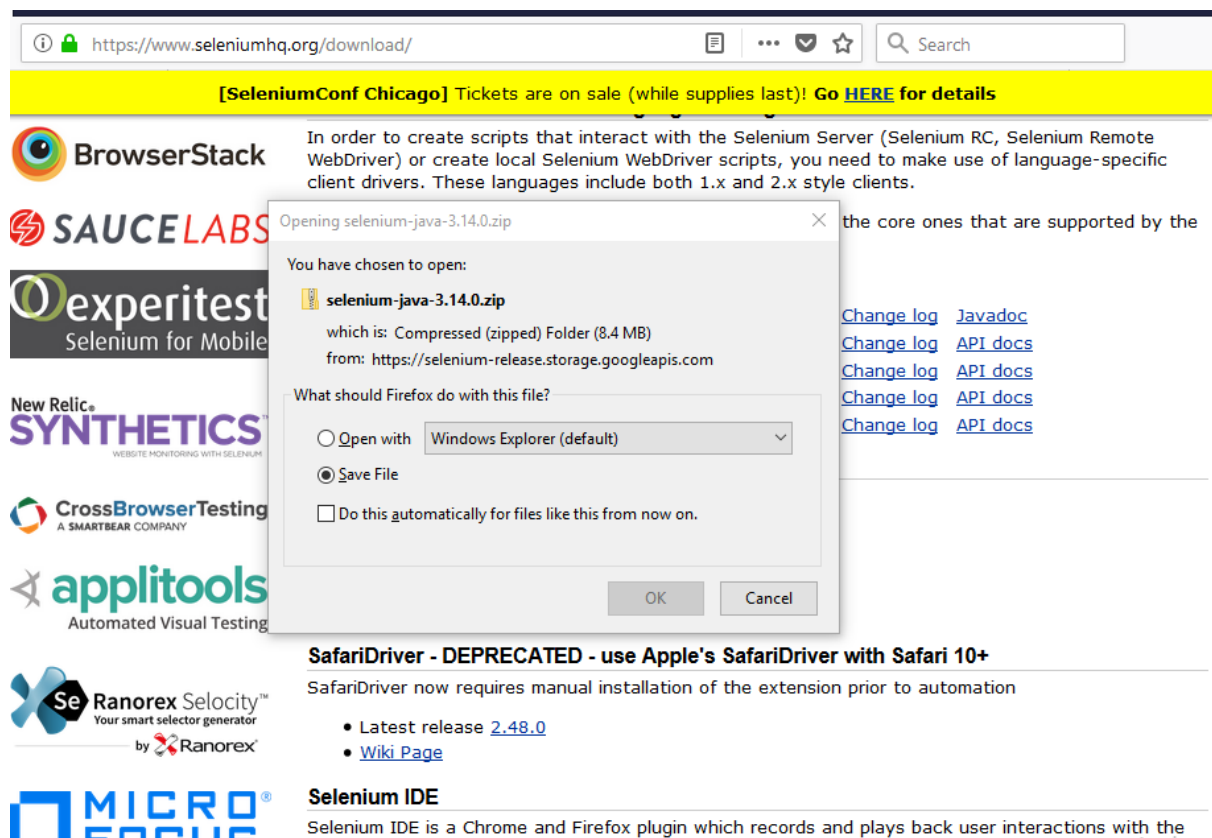
Navigate to the following website,

<https://docs.seleniumhq.org/download/>

Click on the download link of JAVA.

Language	Client Version	Release Date	Download	Change log	Javadoc
Java	3.14.0	2018-08-02	Download	Change log	Javadoc

You will get the below pop up.



Program to handle the file download pop up using AUTOIT?

Below is the script to perform click operation on the "OK" button on file download pop up.

- Press the Key board button "TAB" 3 times, it will select the "OK" button and click on "ENTER" button. It will start downloading the file.
- Below is the AUTOIT script.
- Write it in SciTE, compile and generate the .exe file. Call the .exe from the selenium script.

AUTOIT Script

```
WinWaitActive("Opening selenium-java-3.14.0.zip", "", 5)
Send("{TAB}")
Sleep(2000)
Send("{TAB}")
Sleep(2000)
Send("{TAB}")
Sleep(2000)
Send("{ENTER}")
Sleep(15000)
```

```
import java.io.File;
import java.io.IOException;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class File_download_firefox_using_autoIT {
    public static void main(String[] args) throws IOException, InterruptedException {
        String FileName = "D:/selenium-server-standalone-3.14.0.jar";
        System.setProperty("webdriver.gecko.driver", "D:/Softwares/Drivers/geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.manage().window().maximize();
        driver.get("http://docs.seleniumhq.org/");
        WebElement downloadTab = driver.findElement(By.xpath("//a[text()='Download']"));
        downloadTab.click();
        WebElement javaDownloadLink = driver.findElement(By.xpath("//td[text()='Java']/../a[text()='Download']"));
        javaDownloadLink.click();
        String command = "C:\\Users\\PriyaPramod\\Desktop\\FileUpload\\File_download_firefox.exe";
        Runtime.getRuntime().exec(command);

        File file = new File(FileName);
        if(file.isFile())
        {
            System.out.println("File is downloaded");
        }
        else
        {
            System.out.println("File is not downloaded");
        }
        driver.close();
    }
}
```

Using the ROBOT class

Program to handle file download pop up using the ROBOT class.

```
package popup_file_download;

import java.awt.AWTException;
import java.awt.Robot;
import java.awt.event.KeyEvent;
import java.io.IOException;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class File_download_popUp_using_Robot {
    public static void main(String[] args) throws IOException, InterruptedException, AWTException {

        System.setProperty("webdriver.gecko.driver", "D:/Softwares/Drivers/geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.manage().window().maximize();
        driver.get("http://docs.seleniumhq.org/");
        WebElement downloadTab = driver.findElement(By.xpath("//a[text()='Download']"));
        downloadTab.click();
        WebElement javaDownloadLink = driver.findElement(By.xpath("//td[text()='Java']/../a[text()='Download']"));
        javaDownloadLink.click();
        Thread.sleep(5000);
        Robot robot = new Robot();
        robot.keyPress(KeyEvent.VK_TAB);
        robot.keyRelease(KeyEvent.VK_TAB);
        Thread.sleep(2000);
        robot.keyPress(KeyEvent.VK_TAB);
        robot.keyRelease(KeyEvent.VK_TAB);
        Thread.sleep(2000);
        robot.keyPress(KeyEvent.VK_TAB);
        robot.keyRelease(KeyEvent.VK_TAB);
        Thread.sleep(2000);
        robot.keyPress(KeyEvent.VK_ENTER);

        driver.close();
    }
}
```

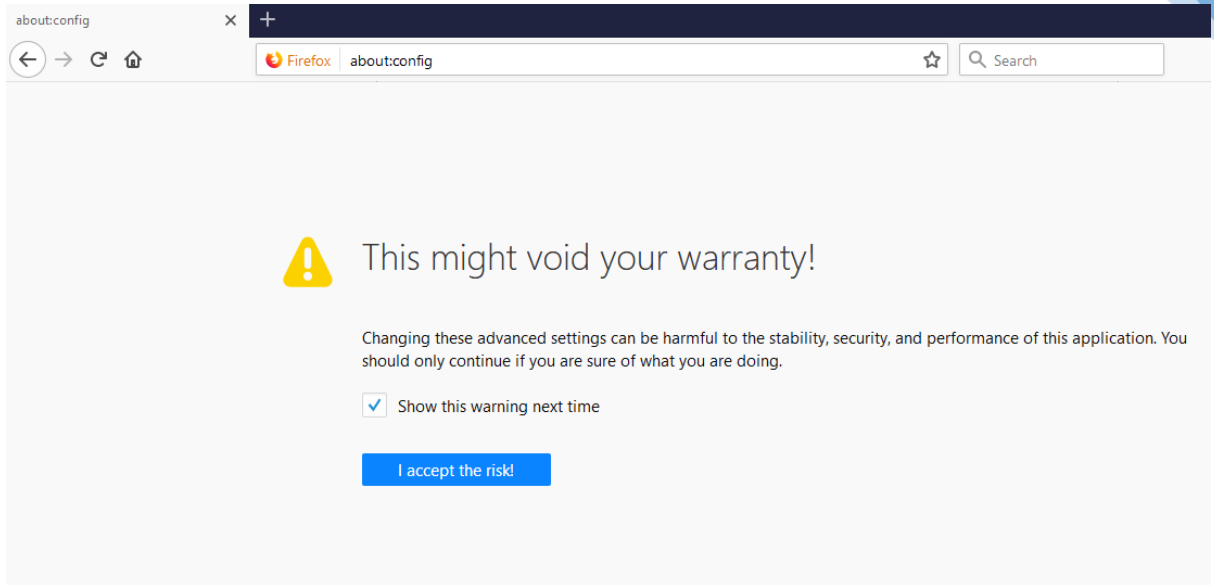
Activat
Get it

By changing the preference or settings of the browser.

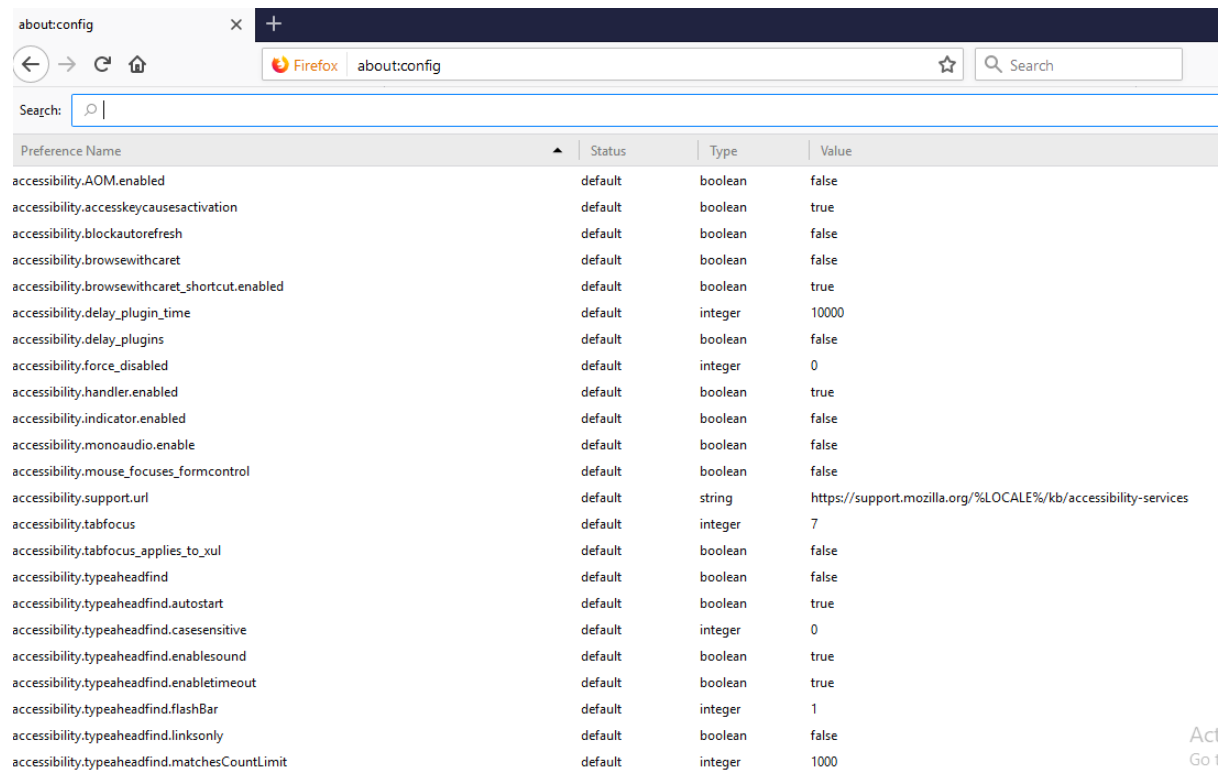
- Webdriver can't perform any operation on file download pop up. We handled using the above two ways.
- In order to handle the file download pop up, we can change the settings or preference of the browser, using the "**FIREFOX OPTIONS**" class.
- Using "**FIREFOX OPTIONS**" class, we can change the behavior of the Firefox browser. So that, we can make Firefox browser to download the file directly without throwing the file download pop up.
- In order to do that, we need to change the Firefox profile. Which takes MIME type.

What is MIME?

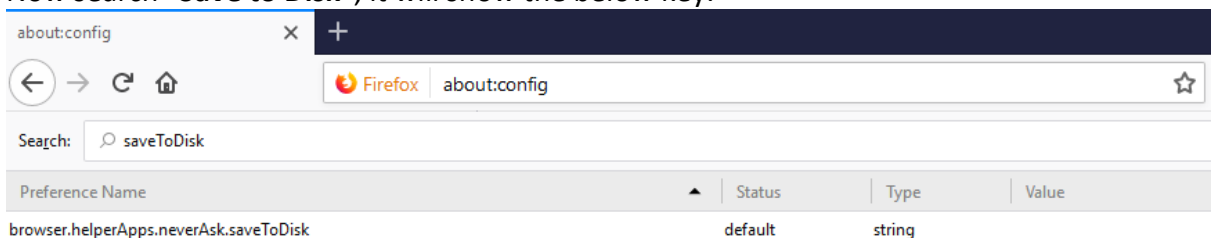
- MIME stands for "**Multipurpose Internet Mail Extension**".
- A MIME type is a label used to identify a type of data. It is used so software can know how to handle the data. It serves the same purpose on the Internet.
- In order to change the settings or preference of the browser, we need to pass the Firefox profile Keys and Values.
- In order to see the setting manually.
- Open the Firefox browser. And enter "**about: config**" on the address bar.



- Click on the **"I accept the risk!"**
- Firefox will show all the keys now.



- Now search **"Save to Disk"**, it will show the below key.



- This is the key, we need to use to change the Firefox behavior to download the file without showing the pop up.

Program to download the file using Firefox Options class?

- Create Object of Firefox Options.
- Call the method “addPreference”, which takes two arguments, “Key” & “Value”.
- Key -> we need to pass the above key to tell the Firefox browser to download the file directly to the driver.
- Value -> here, we need to specify which type of application file we are downloading using the below format.

“application/file type”

Example: if you are downloading zip file, we should mention like below,

“application/zip”

Note: if you pass both these arguments to the addPreference, than Firefox browser will never show the pop up whenever you download zip file.

Below is the code:

```
package popup_file_download;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxOptions;

public class File_download_using_Profile {
    public static void main(String[] args) {

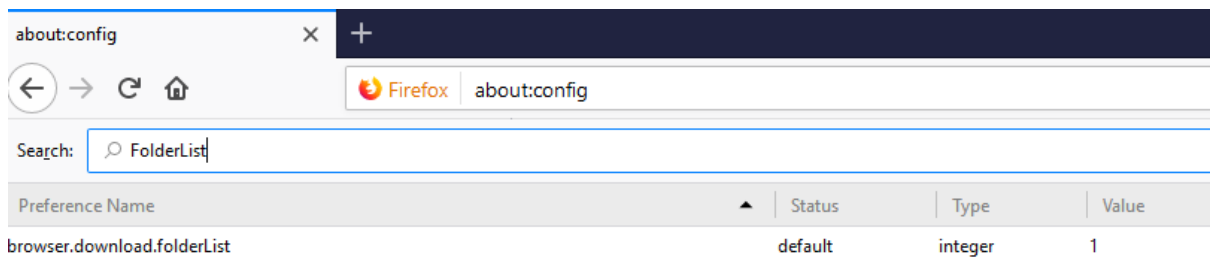
        FirefoxOptions options = new FirefoxOptions();
        options.addPreference("browser.helperApps.neverAsk.saveToDisk", "application/zip");

        System.setProperty("webdriver.gecko.driver", "D:/Softwares/Drivers/geckodriver.exe");
        WebDriver driver = new FirefoxDriver(options);

        driver.manage().window().maximize();
        driver.get("http://docs.seleniumhq.org/");
        driver.findElement(By.xpath("//a[text()='Download']")).click();
        driver.findElement(By.xpath("//td[text()='Java']/../a[text()='Download']")).click();
    }
}
```

Consider, instead of downloading the file to download folder, you wish to download file into the “**Desktop**”, than you need to add one more key to the above code to tell Firefox browser to download the file to desktop.

- To do that, get the below key from Firefox browser.



- It takes the Integer as a value.
- It has a default value as "1".
- 3 values we can pass as a value to the above key. 0 or 1 or 2.

0 -> Downloads the file to desktop

1 -> Downloads the file to download folder

2 -> Downloads the file to specified folder.

Below is the program to download the file to Desktop

```
package popup_file_download;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxOptions;

public class File_download_using_Profile {
    public static void main(String[] args) {

        FirefoxOptions options = new FirefoxOptions();
        options.addPreference("browser.helperApps.neverAsk.saveToDisk", "application/zip");
        options.addPreference("browser.download.folderList", 0);

        System.setProperty("webdriver.gecko.driver", "D:/Softwares/Drivers/geckodriver.exe");
        WebDriver driver = new FirefoxDriver(options);

        driver.manage().window().maximize();
        driver.get("http://docs.seleniumhq.org/");
        driver.findElement(By.xpath("//a[text()='Download']")).click();
        driver.findElement(By.xpath("//td[text()='Java']/../a[text()='Download']")).click();
    }
}
```

Now, if you want to download the file to specified directory. Add one more preference to the above code. By assigning the value "2" to "folderList" preference.

Below is the code to download the file to specified directory.

```
package popup_file_download;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxOptions;

public class File_download_using_Profile {
    public static void main(String[] args) {

        FirefoxOptions options = new FirefoxOptions();
        options.addPreference("browser.helperApps.neverAsk.saveToDisk", "application/zip");
        options.addPreference("browser.download.folderList", 2);
        options.addPreference("browser.download.dir", "D:\\");

        System.setProperty("webdriver.gecko.driver", "D:/Softwares/Drivers/geckodriver.exe");
        WebDriver driver = new FirefoxDriver(options);

        driver.manage().window().maximize();
        driver.get("http://docs.seleniumhq.org/");
        driver.findElement(By.xpath("//a[text()='Download']")).click();
        driver.findElement(By.xpath("//td[text()='Java']/../a[text()='Download']")).click();
    }
}
```

Handling In **Chrome Browser**:

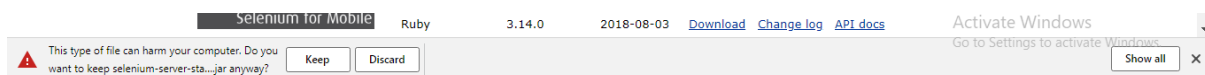
Step 1: Navigate to the following web site.

Step 2: Navigate to the following website,

<https://docs.seleniumhq.org/download/>

Click on the 3.14.0 link.

Step 3: Once after clicking on the download link, you will get the below download pop up.



Note: Now, this is the download pop up, which we need to handle.

To handle this, we have ways to handle it.

1. Using AutoIT
2. Using Robot class
3. Using Chrome Options class

Using AutoIT.

- In order to perform any operation on the download pop up. We need to identify the elements.
- But in this case, AutoIT window info tool will fail to identify the elements.
- So, to handle this. We use the keyboard buttons to select the buttons like "Keep" or "Discard". And in will send all these keyboard buttons using AUTOIT.

Below is the script to perform click operation on the "Keep" button.

- Use Key board button, "F6" to highlight the download button. Need to press F6, three times to highlight the file download pop up.
- After file download highlighted, press "TAB", it will highlight the "KEEP" button.
- If you want to keep the downloaded file. Hit "ENTER"
- Or else press "TAB" again and hit "ENTER" to "DISCARD"

Below is the AUTOIT script.

- Write it in SciTE, compile and generate the .exe file. Call the .exe from the selenium script.

```
WinWaitActive("Downloads - Google Chrome", "", 5)
Send("{F6}")
Sleep(2000)
Send("{F6}")
Sleep(2000)
Send("{F6}")
Sleep(2000)
Send("{TAB}")
Sleep(2000)
Send("{ENTER}")
Sleep(5000)
```

Below is the complete Selenium script to click "Keep" button on the file download pop up.

```
package popup_file_download;

import java.io.IOException;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Chrome_file_download_using_autoIT {
    public static void main(String[] args) throws IOException {
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://docs.seleniumhq.org/");

        driver.findElement(By.xpath("//a[text()='Download']")).click();
        driver.findElement(By.xpath("//a[text()='3.14.0']")).click();

        String command = "C:\\Users\\PriyaPramod\\Desktop\\FileUpload\\File_download.exe";
        Runtime.getRuntime().exec(command);
    }
}
```

Write a script to download and validate if the file has been downloaded or not?

```

import java.io.File;
import java.io.IOException;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Chrome_file_download_using_autoIT {
    public static void main(String[] args) throws IOException {
        String FileName = "C:\\Users\\PriyaPramod\\Downloads\\selenium-server-standalone-3.14.0.jar";
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://docs.seleniumhq.org/");
        driver.findElement(By.xpath("//a[text()='Download']")).click();
        driver.findElement(By.xpath("//a[text()='3.14.0']")).click();

        String command = "C:\\Users\\PriyaPramod\\Desktop\\FileUpload\\File_download.exe";
        Runtime.getRuntime().exec(command);

        File file = new File(FileName);
        if (file.isFile()) {
            System.out.println("File has been downloaded succussfully");
        } else {
            System.out.println("File is not downloaded");
        }
        driver.close();
    }
}

```

Activate Windows
Go to Settings to activate Windows.

Using ROBOT class

- Another way to handle the file download pop up in chrome browser is using the robot class.

Script to handle the file download pop up using the robot class.

```

import java.io.File;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Chrome_file_download_Robot {
    public static void main(String[] args) throws AWTException, InterruptedException {
        String FileName = "C:\\Users\\PriyaPramod\\Downloads\\selenium-server-standalone-3.14.0.jar";
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://docs.seleniumhq.org/");
        driver.findElement(By.xpath("//a[text()='Download']")).click();
        driver.findElement(By.xpath("//a[text()='3.14.0']")).click();

        Thread.sleep(10000);
        Robot robot = new Robot();
        robot.keyPress(KeyEvent.VK_F6);
        robot.keyRelease(KeyEvent.VK_F6); Thread.sleep(1000);
        robot.keyPress(KeyEvent.VK_F6);
        robot.keyRelease(KeyEvent.VK_F6); Thread.sleep(1000);
        robot.keyPress(KeyEvent.VK_F6);
        robot.keyRelease(KeyEvent.VK_F6); Thread.sleep(1000);

        robot.keyPress(KeyEvent.VK_TAB);
        robot.keyRelease(KeyEvent.VK_TAB); Thread.sleep(1000);
        robot.keyPress(KeyEvent.VK_ENTER);
        robot.keyRelease(KeyEvent.VK_ENTER); Thread.sleep(1000);

        File file = new File(FileName);
        if (file.isFile()) {
            System.out.println("PASS: File has been downloaded succussfully");
        } else {
            System.out.println("FAIL: File is not downloaded");
        }
        driver.close();
    }
}

```

Using Chrome Options Class.

Pramod K S

- This is one of the best way to download the file.
- By changing the behavior of the browser, we can download the file and as well we can download the file to the specified directory.

Below is the code to download the file by changing chrome browser. And downloading the file to specified directory.

```
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;

public class Chrome_file_download_using_options {
    public static void main(String[] args) throws IOException, InterruptedException {

        String downloadFilepath = "D:/";
        String FileName = "D:/selenium-server-standalone-3.11.0.jar";

        Map<String, Object> prefs = new HashMap<String, Object>();
        prefs.put("profile.default_content_settings.popups", 0);
        prefs.put("download.default_directory", downloadFilepath);
        prefs.put("safebrowsing.enabled", true);
        ChromeOptions options = new ChromeOptions();
        options.setExperimentalOption("prefs", prefs);

        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver(options);
        driver.manage().window().maximize();
        driver.get("http://docs.seleniumhq.org/");
        driver.findElement(By.xpath("//a[text()='Download']")).click();
        driver.findElement(By.xpath("//td[text()='Java']/../a[text()='Download']")).click();

        File file = new File(FileName);
        if (file.isFile()) {
            System.out.println("File is downloaded");
        } else {
            System.out.println("File is not downloaded");
        }
        driver.close();
    }
}
```