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,
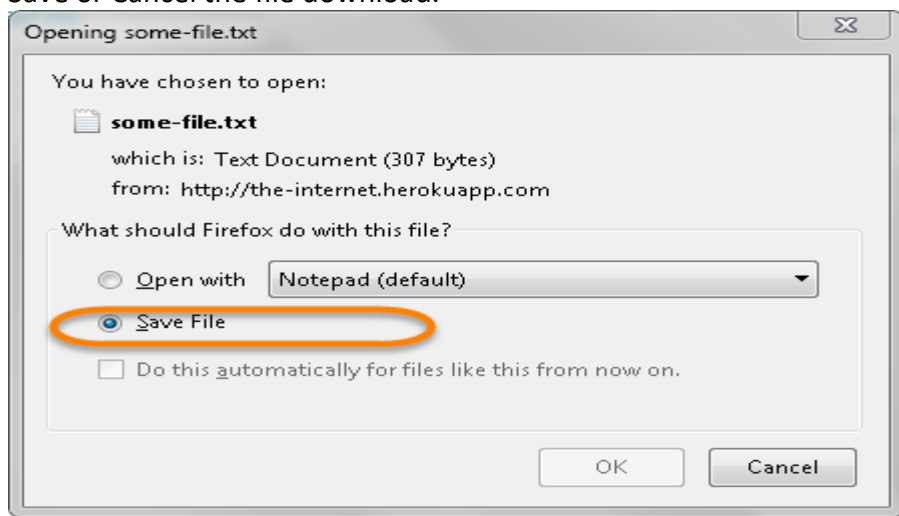 - ➢ We cannot inspect the file download pop up.
    Note: Once click on the link of the file, one dialog box or confirmation window will appears and ask to save the file. If you try to inspect the element of the window, you will notice that it is not the browser HTML window but a system window, which 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 handle file download pop up, we have 2 different ways.

Pramod K S

1. Using AutoIT
2. Using Firefox profile 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.
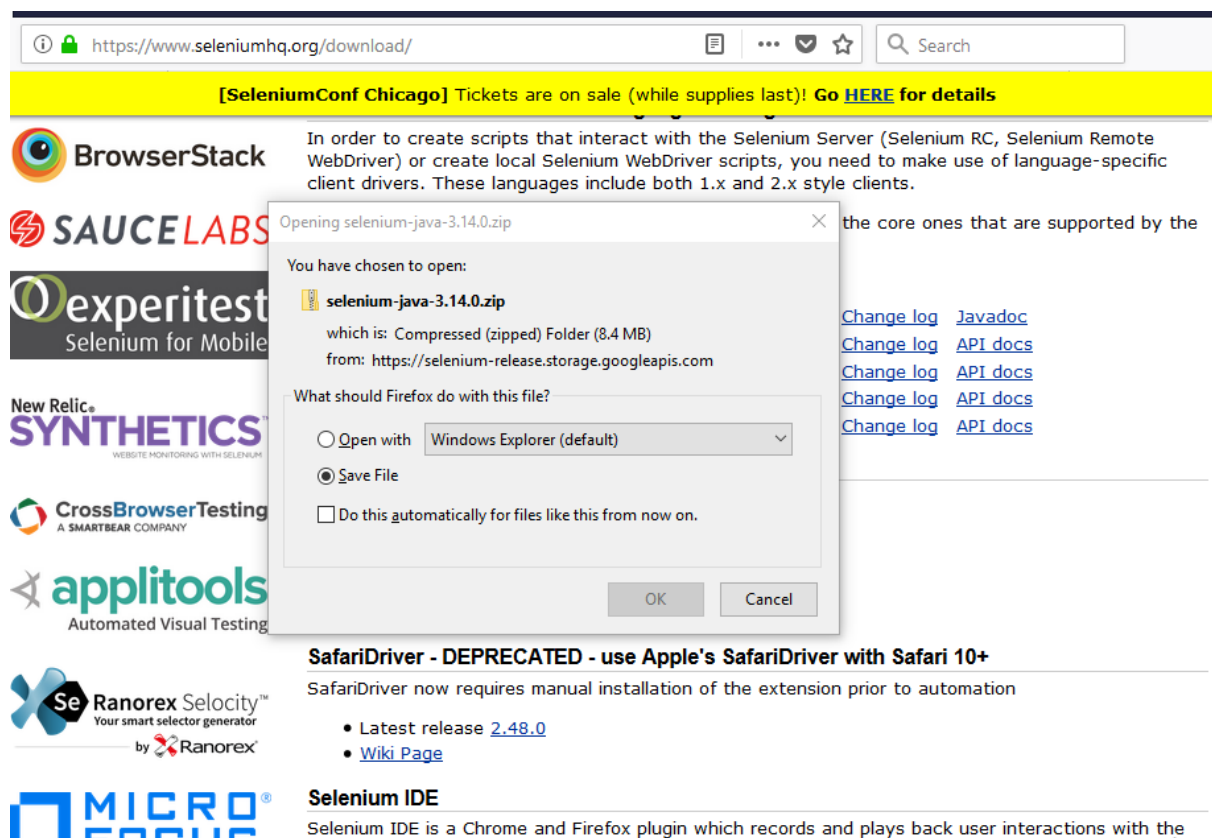
Navigate to the following website,

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

Click on the download link of JAVA.

Language	Client Version	Release Date			
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

```
from selenium import webdriver
import autoit
from pathlib import Path

file_path = Path("C:/Users/PriyaPramod/Downloads/selenium-java-3.14.0.zip")

driver = webdriver.Firefox()
driver.implicitly_wait(30)
driver.get("http://docs.seleniumhq.org/")

download_tab = driver.find_element_by_xpath("//a[text()='Download']")
download_tab.click()
java_download_link = driver.find_element_by_xpath("//td[text()='Java']/..//a[text()='Download']")
java_download_link.click()

autoit.win_wait_active("Opening selenium-java-3.141.59.zip")
autoit.send("{TAB}")
autoit.send("{TAB}")
autoit.send("{TAB}")
autoit.send("{TAB}")
autoit.send("{ENTER}")

driver.close()
```

Write a program to verify the file has been downloaded or not

```
from selenium import webdriver
import autoit
from pathlib import Path

file_path = Path("C:/Users/PriyaPramod/Downloads/selenium-java-3.14.0.zip")

driver = webdriver.Firefox()
driver.implicitly_wait(30)
driver.get("http://docs.seleniumhq.org/")

download_tab = driver.find_element_by_xpath("//a[text()='Download']")
download_tab.click()
java_download_link = driver.find_element_by_xpath("//td[text()='Java']/..//a[text()='Download']")
java_download_link.click()

autoit.win_wait_active("Opening selenium-java-3.141.59.zip")
autoit.send("{TAB}")
autoit.send("{TAB}")
autoit.send("{TAB}")
autoit.send("{TAB}")
autoit.send("{ENTER}")

if file_path.is_file():
    print("True: File is down-loaded")
else:
    print("False: File is not down-loaded")

driver.close()
```
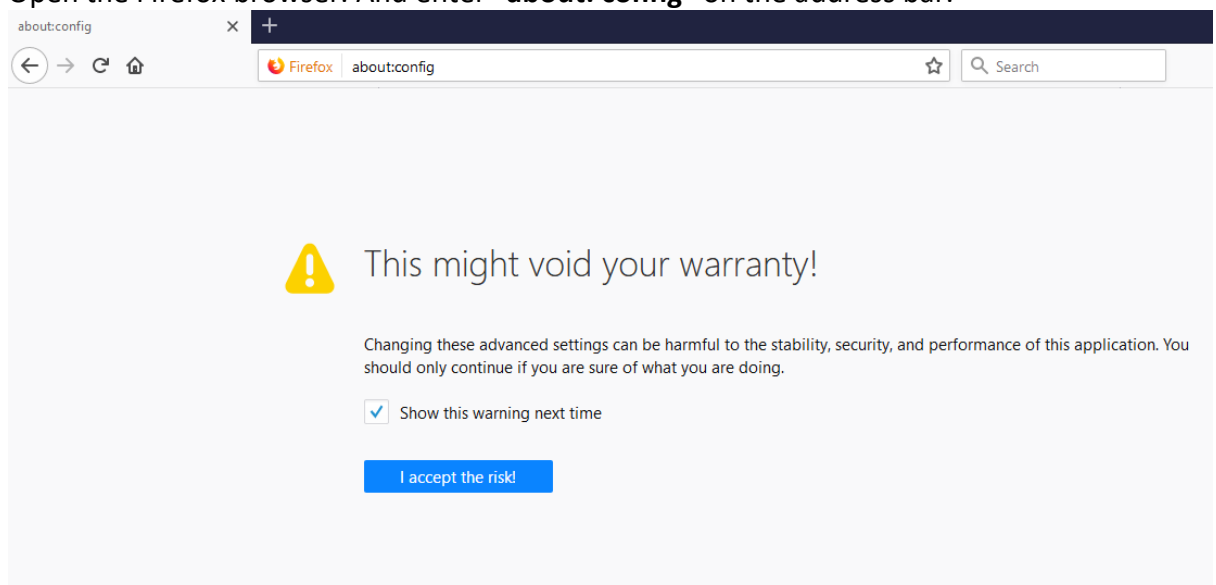
Pramod K S

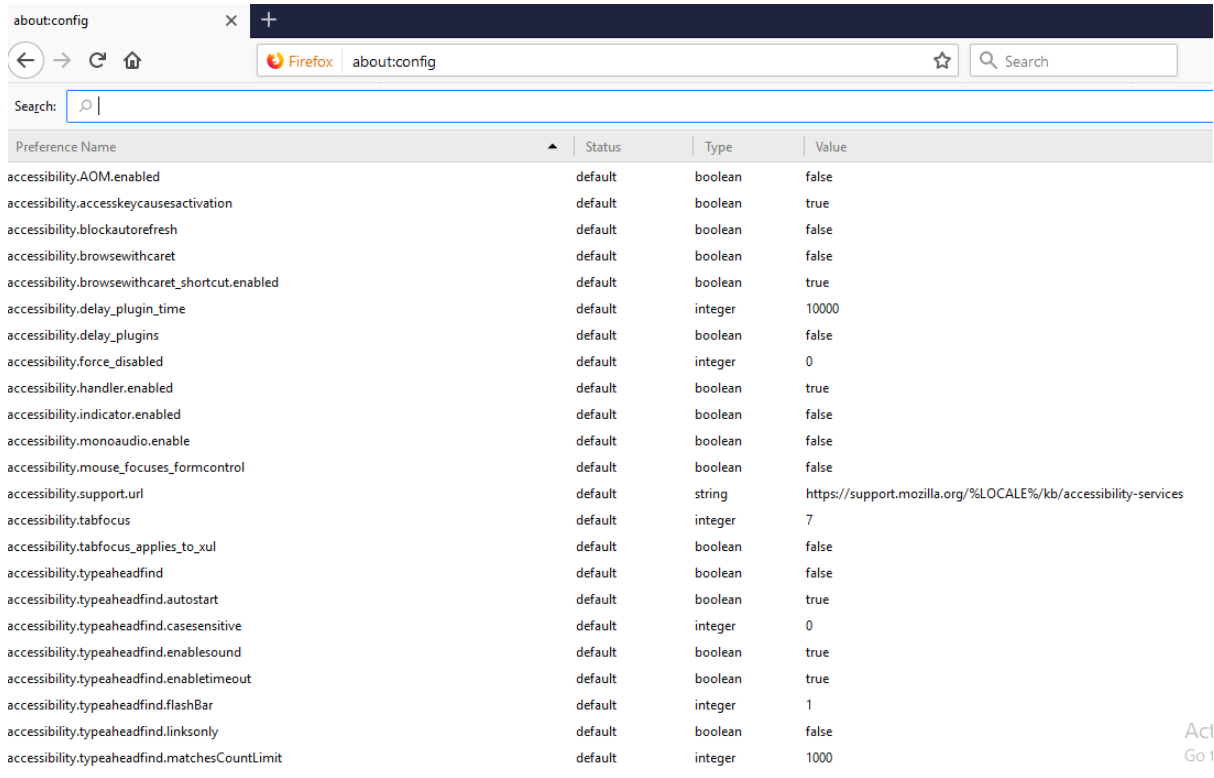**By changing the preference or settings of the browser.**

- 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 PROFILE**" class.
- Using "**FIREFOX PROFILE**" 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.
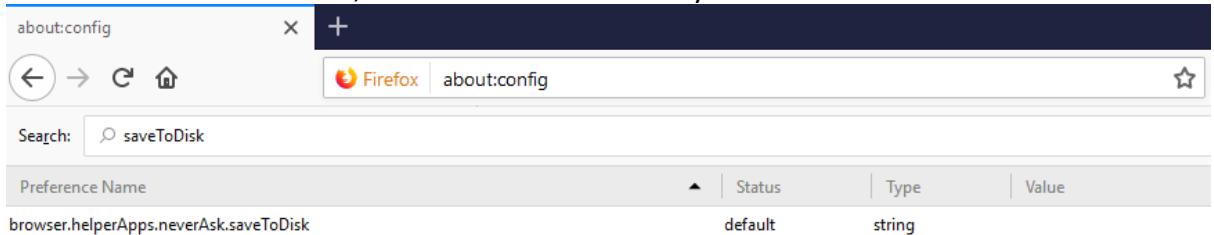
What is MIME?

- 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 Profile.
- 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.

Pramod K S

Below is the code:

```python
from selenium import webdriver

ffProfile = webdriver.FirefoxProfile()
ffProfile.set_preference("browser.helperApps.neverAsk.saveToDisk", "application/zip")

driver = webdriver.Firefox(ffProfile)
driver.get("https://docs.seleniumhq.org/download/")
driver.implicitly_wait(100)

download_link = driver.find_element_by_xpath("//td[text()='Java']/..//a[text()='Download']")
download_link.click()

driver.close()
```
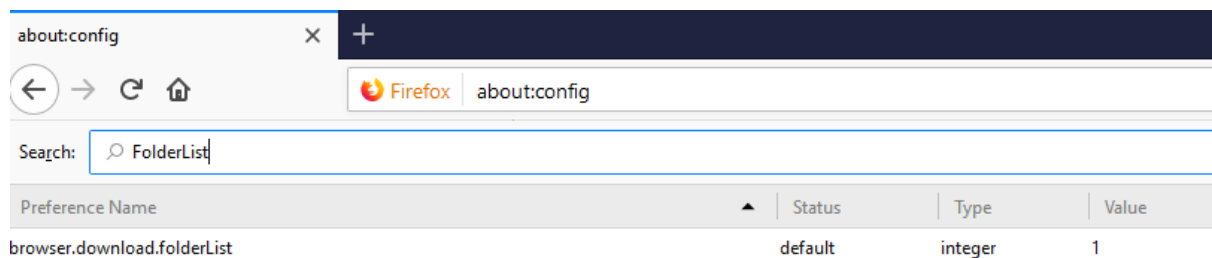
Consider, instead of downloading the file to download folder, you wish to download file into the "**Desktop**", than you need to add one more key to the above code to tell Firefox browser to download the file to desktop.

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

about:config	×	+				
← → C ⌂		🦊 Firefox	about:config			
Search: 🔍 FolderList						
Preference Name			▲	**Status**	**Type**	**Value**
browser.download.folderList				default	integer	1

- 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

```python
from selenium import webdriver

ffProfile = webdriver.FirefoxProfile()
ffProfile.set_preference("browser.helperApps.neverAsk.saveToDisk", "application/zip")
ffProfile.set_preference("browser.download.folderList", 0)

driver = webdriver.Firefox(ffProfile)
driver.get("https://docs.seleniumhq.org/download/")
driver.implicitly_wait(100)

download_link = driver.find_element_by_xpath("//td[text()='Java']/..//a[text()='Download']")
download_link.click()

driver.close()
```

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.

Pramod K S

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

```python
from selenium import webdriver

ffProfile = webdriver.FirefoxProfile()
ffProfile.set_preference("browser.helperApps.neverAsk.saveToDisk", "application/zip")
ffProfile.set_preference("browser.download.folderList", 2)
ffProfile.set_preference("browser.download.dir", "D:\\")

driver = webdriver.Firefox(ffProfile)
driver.get("https://docs.seleniumhq.org/download/")
driver.implicitly_wait(100)

download_link = driver.find_element_by_xpath("//td[text()='Java']/..//a[text()='Download']")
download_link.click()

driver.close()
```

MIME type for other type of files.

Extension	Kind of document	MIME Type
.aac	AAC audio	audio/aac
.abw	AbiWord document	application/x-abiword
.arc	Archive document (multiple files embedded)	application/x-freearc
.avi	AVI: Audio Video Interleave	video/x-msvideo
.azw	Amazon Kindle eBook format	application/vnd.amazon.ebook
.bin	Any kind of binary data	application/octet-stream
.bmp	Windows OS/2 Bitmap Graphics	image/bmp
.bz	BZip archive	application/x-bzip
.bz2	BZip2 archive	application/x-bzip2
.csh	C-Shell script	application/x-csh
.css	Cascading Style Sheets (CSS)	text/css
.csv	Comma-separated values (CSV)	text/csv
.doc	Microsoft Word	application/msword
.docx	Microsoft Word (OpenXML)	application/vnd.openxmlformats-officedocument.wordprocessingml.d
.eot	MS Embedded OpenType fonts	application/vnd.ms-fontobject
.epub	Electronic publication (EPUB)	application/epub+zip
.gif	Graphics Interchange Format (GIF)	image/gif
.htm .html	HyperText Markup Language (HTML)	text/html
.ico	Icon format	image/vnd.microsoft.icon

Pramod K S

Extension	Kind of document	MIME Type
`.ics`	iCalendar format	`text/calendar`
`.jar`	Java Archive (JAR)	`application/java-archive`
`.jpeg` `.jpg`	JPEG images	`image/jpeg`
`.js`	JavaScript	`text/javascript`
`.json`	JSON format	`application/json`
`.jsonld`	JSON-LD format	`application/ld+json`
`.mid` `.midi`	Musical Instrument Digital Interface (MIDI)	`audio/midi audio/x-midi`
`.mjs`	JavaScript module	`text/javascript`
`.mp3`	MP3 audio	`audio/mpeg`
`.mpeg`	MPEG Video	`video/mpeg`
`.mpkg`	Apple Installer Package	`Application /vnd.apple.installer+xml`
`.odp`	OpenDocument presentation document	`Application /vnd.oasis.opendocument .presentation`
`.ods`	OpenDocument spreadsheet document	`Application /vnd.oasis.opendocument .spreadsheet`
`.odt`	OpenDocument text document	`Application /vnd.oasis.opendocument.text`
`.oga`	OGG audio	`audio/ogg`
`.ogv`	OGG video	`video/ogg`
`.ogx`	OGG	`application/ogg`
`.otf`	OpenType font	`font/otf`
`.png`	Portable Network Graphics	`image/png`
`.pdf`	Adobe Portable Document Format (PDF)	`application/pdf`
`.ppt`	Microsoft PowerPoint	`Application /vnd.ms-powerpoint`
`.pptx`	Microsoft PowerPoint (OpenXML)	`Application /vnd.openxmlformats -officedocument .presentationml.presentation`

Pramod K S

Extension	Kind of document	MIME Type
`.rar`	RAR archive	`application/x-rar-compressed`
`.rtf`	Rich Text Format (RTF)	`application/rtf`
`.sh`	Bourne shell script	`application/x-sh`
`.svg`	Scalable Vector Graphics (SVG)	`image/svg+xml`
`.swf`	Small web format (SWF) or Adobe Flash document	`Application/x-shockwave-flash`
`.tar`	Tape Archive (TAR)	`application/x-tar`
`.tif` `.tiff`	Tagged Image File Format (TIFF)	`image/tiff`
`.ts`	MPEG transport stream	`video/MP2T`
`.ttf`	TrueType Font	`font/ttf`
`.txt`	Text, (generally ASCII or ISO 8859-n)	`text/plain`
`.vsd`	Microsoft Visio	`application/vnd.visio`
`.wav`	Waveform Audio Format	`audio/wav`
`.weba`	WEBM audio	`audio/webm`
`.webm`	WEBM video	`video/webm`
`.webp`	WEBP image	`image/webp`
`.woff`	Web Open Font Format (WOFF)	`font/woff`
`.woff2`	Web Open Font Format (WOFF)	`font/woff2`
`.xhtml`	XHTML	`application/xhtml+xml`
`.xls`	Microsoft Excel	`application/vnd.ms-excel`
`.xlsx`	Microsoft Excel (OpenXML)	`Application/vnd.openxmlformats-officedocument.spreadsheetml.sheet`
`.xml`	XML	`application/xml or text/xml`
`.xul`	XUL	`Application/vnd.mozilla.xul+xml`
`.zip`	ZIP archive	`application/zip`
`.3gp`	3GPP audio/video container	`video/3gpp` `audio/3gpp`

Pramod K S

Extension	Kind of document	MIME Type
.3g2	3GPP2 audio/video container	video/3gpp2 audio/3gpp2
.7z	7-zip archive	application/x-7z-compressed

Pramod K S