

Selenium Java - Tutorial 13- Java alerts, Notifications

This is the next tutorial in selenium-java series. Please go through the previous tutorials before you start this one. In the last tutorial, we learned how to handle dropdowns. In this tutorial we will see how to handle alert popups, notifications!

What you will Learn:

1. Handle Java alerts
2. Handle Notifications
3. Desired capabilities
4. Firefox profile

Handle Java Alerts

Navigate to <http://www.tizag.com/javascriptT/javascriptalert.php> & scroll towards the bottom of the page. You will see 'Display' section having 'Confirmation Alert' button

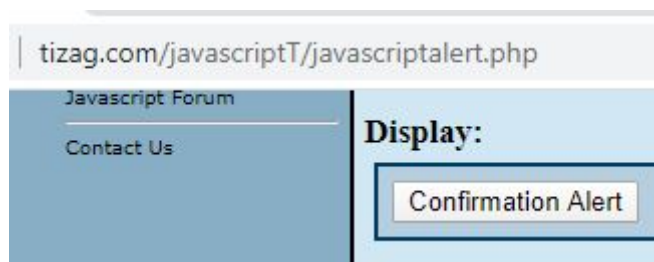


Figure 1

Click 'Confirmation Alert' button. A popup message will appear



Figure 2

Try to right click on this popup alert. You will not be able to do so. The reason being, this popup is a java alert popup and it is not web related. Since it not web related, we cannot inspect this popup by right click. To identify this popup, we will have to switch the driver from webpage to popup using

'Alert' class. However note that you should use this 'Alert' class only if it java alert (viz not able to identify any element), not always.

'Alert' is an interface & you can read the api documentation here. Notice below that this interface has methods such as dismiss(), getText() etc

The screenshot shows a web browser displaying the Selenium Java API documentation for the `Alert` interface. The browser's address bar shows the URL `selenium.dev/selenium/docs/api/java/index.html`. On the left, a sidebar lists various classes and packages, with `org.openqa.selenium.Alert` selected. The main content area displays the `org.openqa.selenium` package and the `Interface Alert`. Below the package name, it states `public interface Alert`. A section titled `Method Summary` contains a table with two columns: `Modifier and Type` and `Method and Description`. The table lists four methods: `accept()`, `dismiss()`, `getText()`, and `sendKeys()`. The `dismiss()` and `getText()` methods are highlighted with blue boxes.

Modifier and Type	Method and Description
void	<code>accept()</code>
void	<code>dismiss()</code>
<code>java.lang.String</code>	<code>getText()</code>
void	<code>sendKeys()</code>

Figure 3

You can click the method to see its description

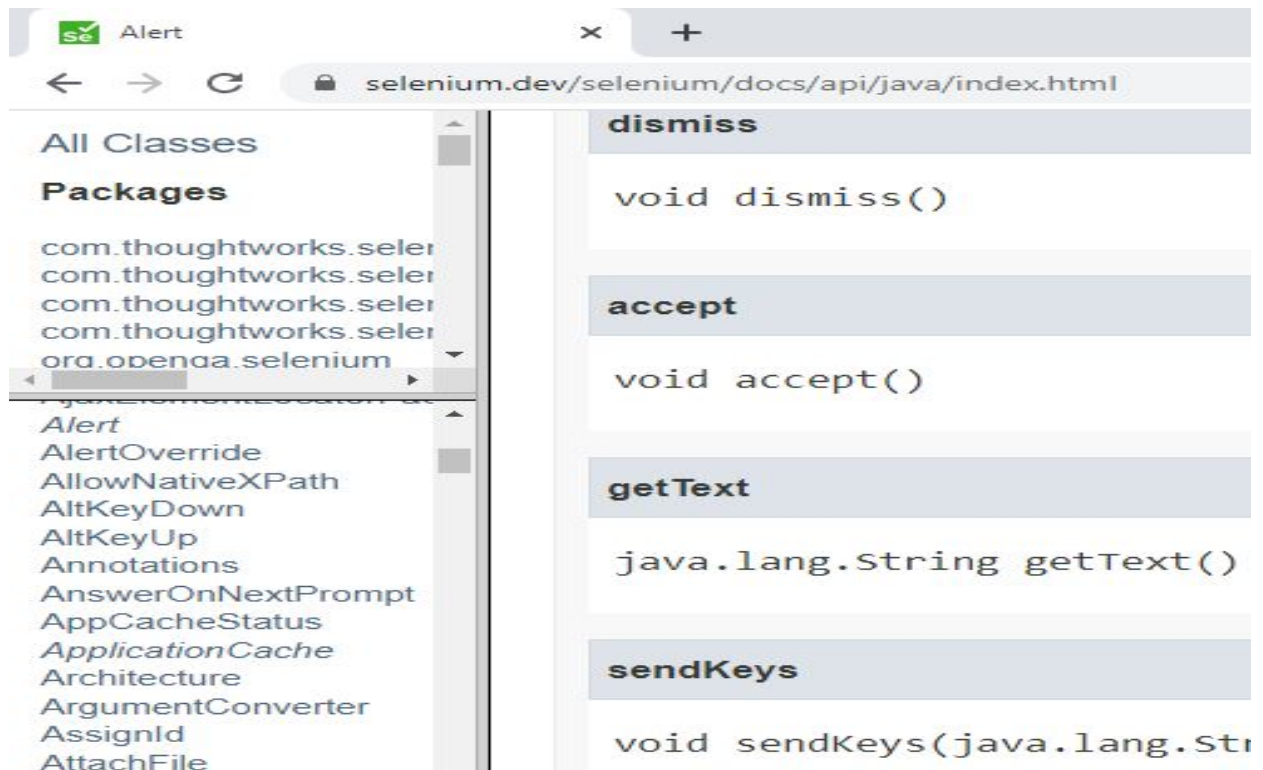


Figure 4

So for java alert popups, we can simply use the method: **switchTo().alert().accept()**
Inspect the 'Confirmation Alert' button

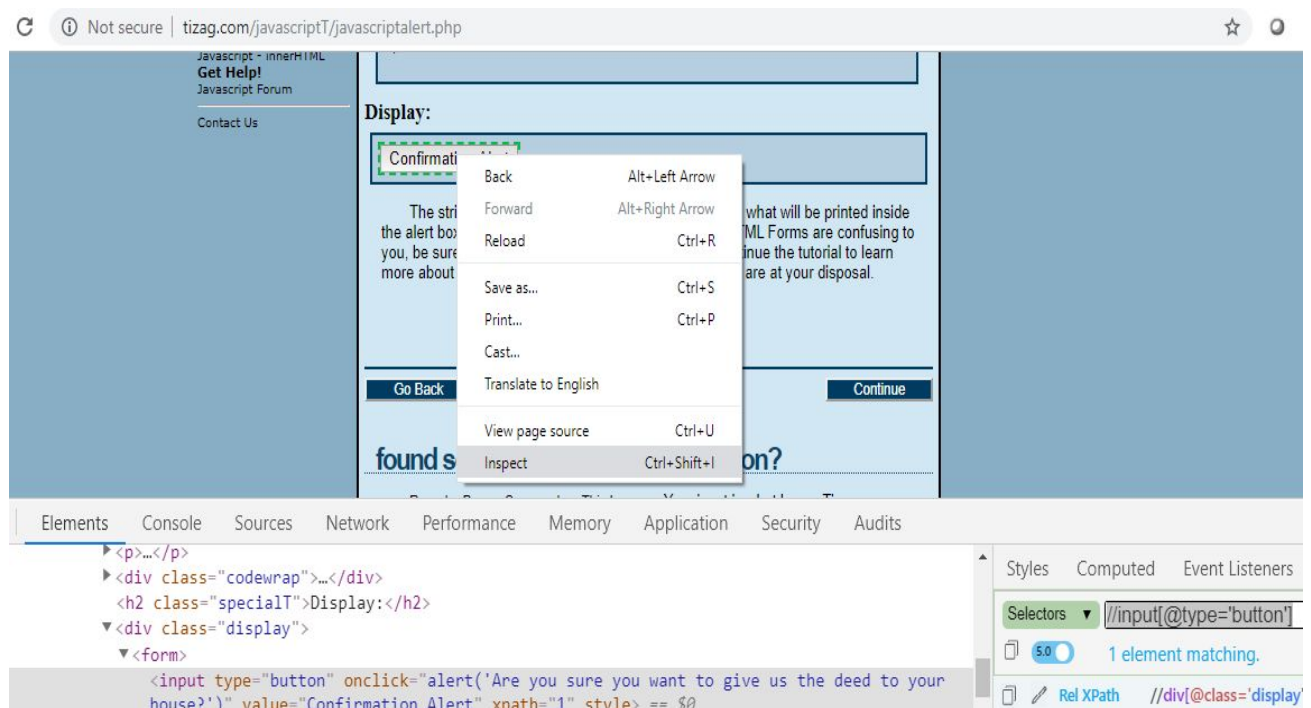


Figure 5

Create new class.

In line#15, we are clicking the 'Confirmation Alert' button.

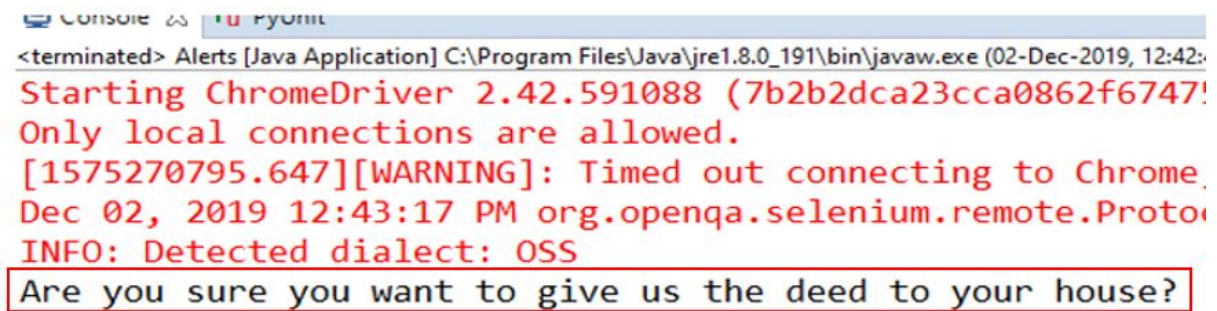
After clicking the button, the popup comes up & we are switching to this alert popup (line#16).

Using `getText()` method, we are printing the text seen in the alert popup message(line#16).
Using `accept()` method, selenium clicks the 'Ok' button on the popup (line#17)

```
11  
12     driver = new ChromeDriver();  
13  
14     driver.get("http://www.tizag.com/javascriptT/javascriptalert.php");  
15     driver.findElement(By.xpath("//input[@type='button']")).click();  
16     System.out.println(driver.switchTo().alert().getText());  
17     driver.switchTo().alert().accept();
```

Figure 6

Run the script, you will see the alert popup's text getting printed in console



The screenshot shows a Java console window with the following output:
<terminated> Alerts [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (02-Dec-2019, 12:42:...) Starting ChromeDriver 2.42.591088 (7b2b2dca23cca0862f6747...) Only local connections are allowed.
[1575270795.647][WARNING]: Timed out connecting to Chrome.
Dec 02, 2019 12:43:17 PM org.openqa.selenium.remote.Protocol...
INFO: Detected dialect: OSS
Are you sure you want to give us the deed to your house?

Figure 7

Navigate to

https://www.wishpond.com/pricing/?utm_source=blog%20overlay&blog_overlay=true&popup

You would see 'Contact Us' button.



wishpond.com/pricing/?utm_source=blog%20overlay&blog_overlay=true&popup

JumpStart Program

Need a hand implementing your marketing strategy?

Through Wishpond's JumpStart Program our team of dedicated experts (marketers, project managers, ad specialists) will work with you to set up a custom marketing strategy made specifically for your business.

Contact Us

Figure 8

Click this button. Web related popup come up. We can spy the objects in this popup & can use normal xpaths. We should not use alert class over here since this is not a java popup.

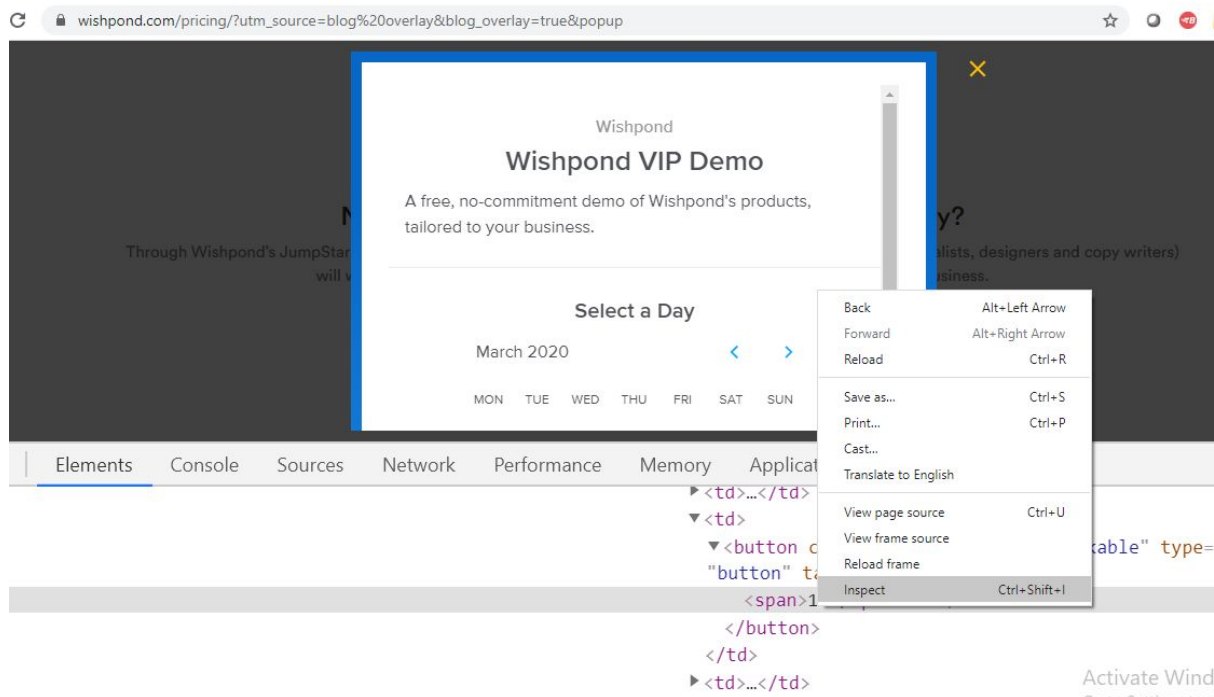


Figure 9

Handle Notifications

In any web application, notifications can appear randomly and as soon as they appear, the driver's focus will shift from the web page and it will shift to the unwanted notification. This will fail your automated script due to element not intractable exception. So we have to implement some mechanism where we can check repeatedly for notifications and if they appear, we should be able to allow/don't allow them and continue with our test execution.

Create new class & launch cleartrip.com

```

4 public class Notifications {
5
6     public static void main(String[] args) {
7         System.setProperty("webdriver.chrome.driver",
8
9         WebDriver driver = null;
10
11         driver = new ChromeDriver();
12
13         driver.get("https://www.cleartrip.com/");
14     }
15 }

```

Figure 10

When you run this script, the below notification would come up. This is NOT a java alert. This is a notifications. These unwanted notifications might be a disturbance when you are running the selenium script. So how to handle these kind of notifications in chrome?

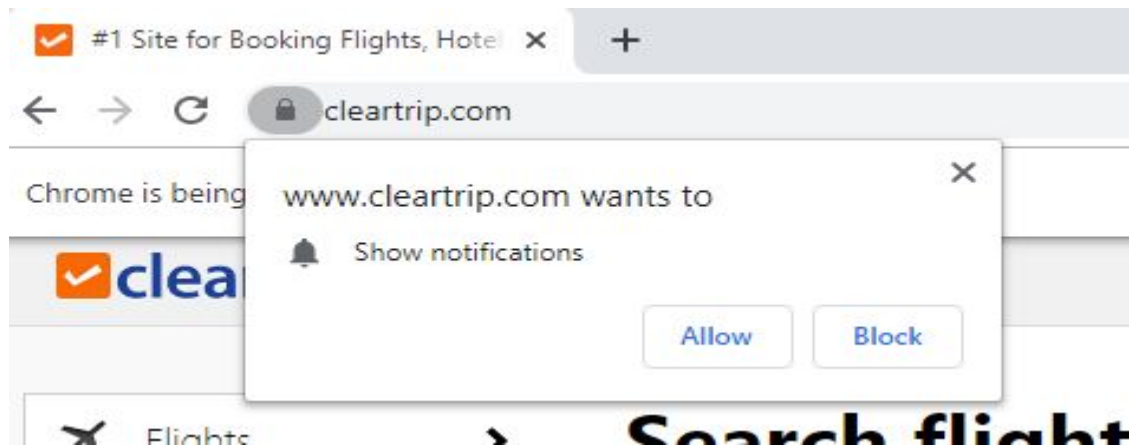


Figure 11

In the api documentation, you can look for 'ChromeOptions' class. This class is used to manage options specific to chrome driver

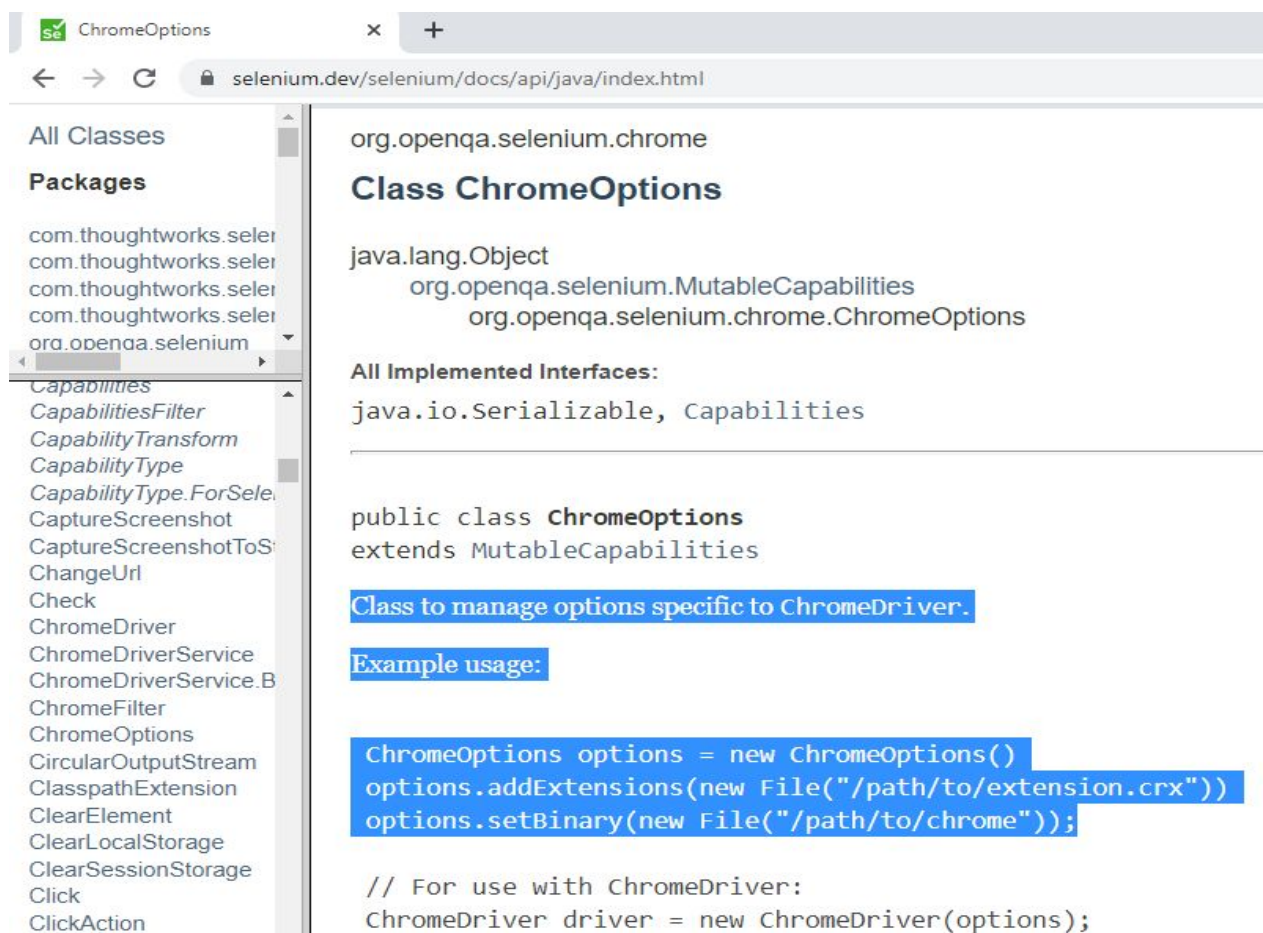


Figure 12

You can add various arguments while starting chrome browser

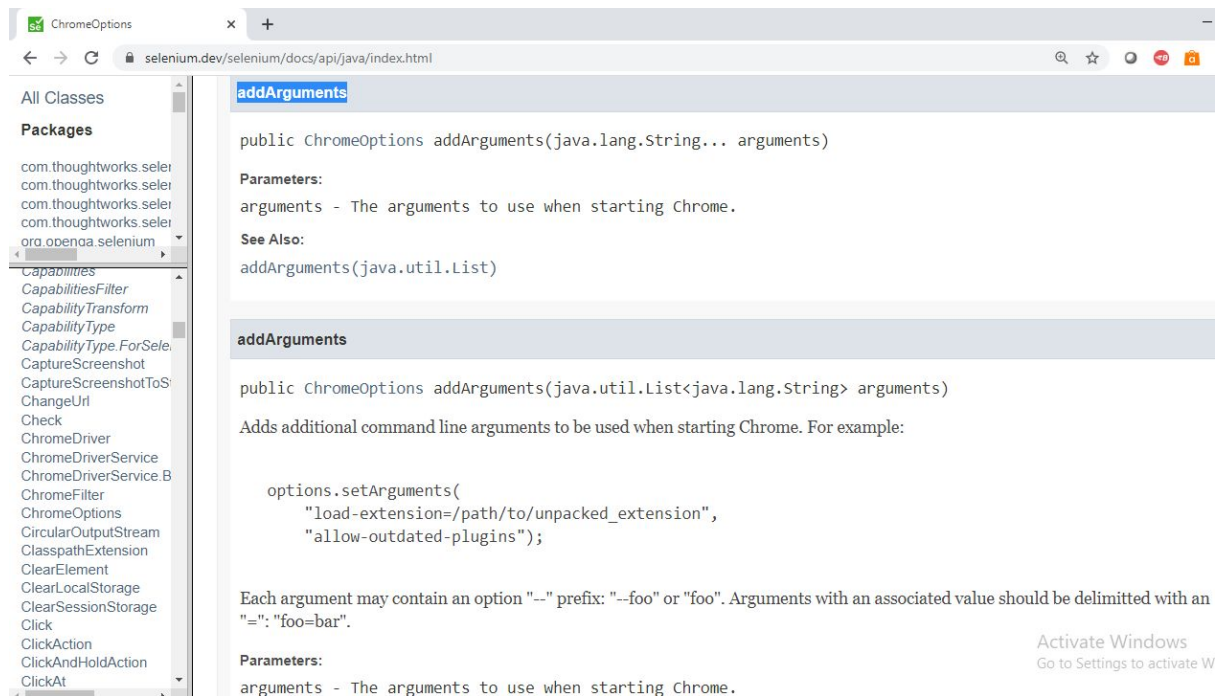


Figure 13

So to handle notifications, add lines 9 & 10. This will disable all the notifications during the launch of chrome browser. Pass the 'ops' reference (that we have created in line#9) into line#16

```

5 public class Notifications {
6
7     public static void main(String[] args) {
8
9         ChromeOptions ops = new ChromeOptions();
10        ops.addArguments("--disable-notifications");
11
12        System.setProperty("webdriver.chrome.driver", "C:\\\\
13
14        WebDriver driver = null;
15
16        driver = new ChromeDriver(ops);
17
18        driver.get("https://www.cleartrip.com/");

```

Figure 14

Now when you run the script, you will not see any notifications, see below

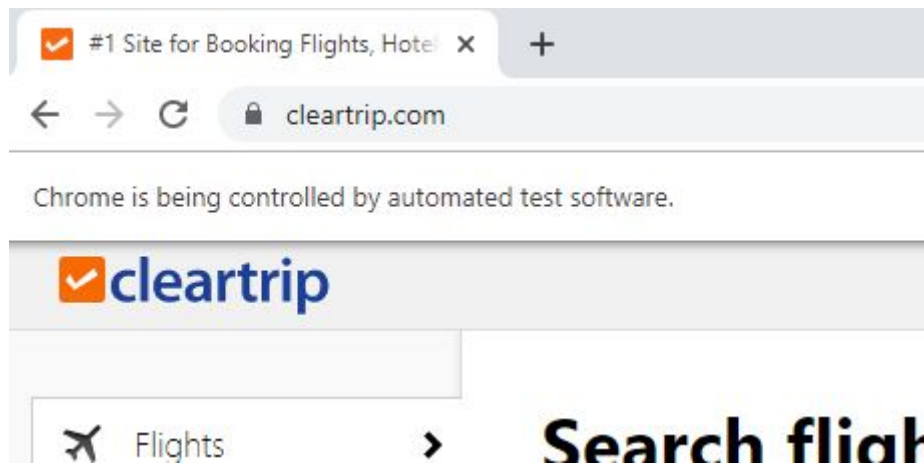


Figure 15

Desired capabilities

You can also handle notifications using DesiredCapabilities class. Below is the standard syntax that you can use.

```
DesiredCapabilities chromeCapabilities = DesiredCapabilities.chrome();
chromeCapabilities.setCapability(CapabilityType.UNEXPECTED_ALERT_BEHAVIOUR,
UnexpectedAlertBehaviour.ACCEPT);
ChromeOptions chromeOptions = new ChromeOptions();
chromeCapabilities.setCapability(ChromeOptions.CAPABILITY, chromeOptions);
ChromDriever driver = new ChromeDriver(chromeCapabilities);
```

You can read about DesiredCapabilities class from api document

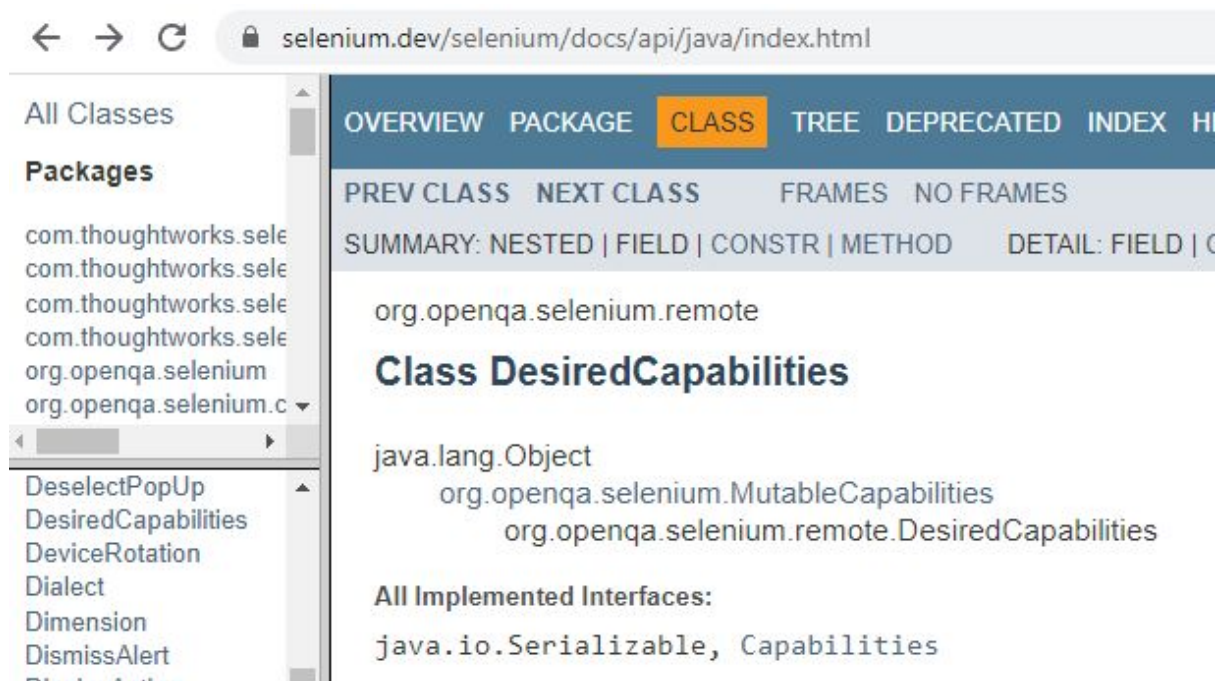


Figure 16

Firefox profile

You can also create a firefox profile and use that profile in your selenium scripts. The advantage of doing this is, whatever bookmarks, settings etc you have done for your chrome browser, the settings

etc are retained when selenium launches browser. If you don't create a profile, selenium will open a brand new session of chrome browser every time. To create firefox profile, type this command in windows: **firefox.exe -p profilemanager**

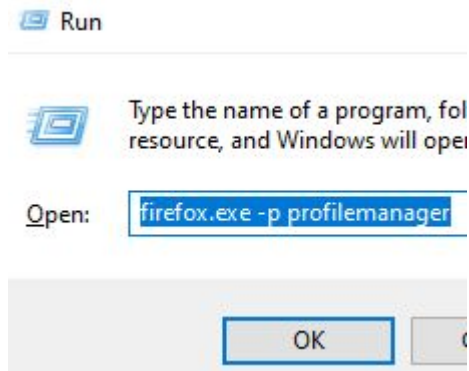


Figure 17

Now click OK, the below comes up

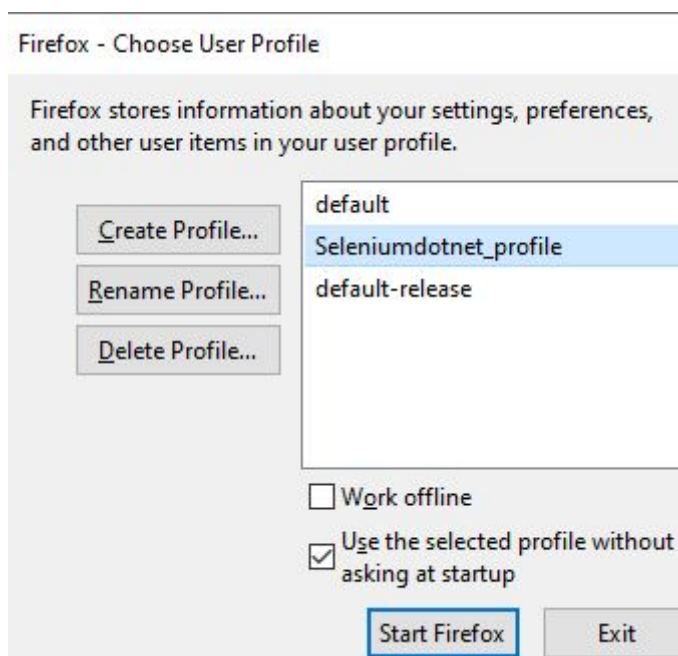


Figure 18

Click 'Create Profile', below comes up

Create Profile Wizard

Welcome to the Create Profile Wizard

Firefox stores information about your settings and preferences in your personal profile.

If you are sharing this copy of Firefox with other users, you can use profiles to keep each user's information separate. To do this, each user should create his or her own profile.

If you are the only person using this copy of Firefox, you must have at least one profile. If you would like, you can create multiple profiles for yourself to store different sets of settings and preferences. For example, you may want to have separate profiles for business and personal use.

To begin creating your profile, click Next.

< Back

Next >

Cancel

Figure 19

Click 'Next', enter the desired profile name you wish to, see below

Create Profile Wizard

Completing the Create Profile Wizard

If you create several profiles you can tell them apart by the profile names. You may use the name provided here or use one of your own.

Enter new profile name:

SeleniumProfile

Your user settings, preferences and other user-related data will be stored in:

C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles
\hs3aafv4.SeleniumProfile

Choose Folder...

Use Default Folder

Click Finish to create this new profile.

< Back

Finish

Cancel

Figure 20

Finish, you will now see the new profile listed

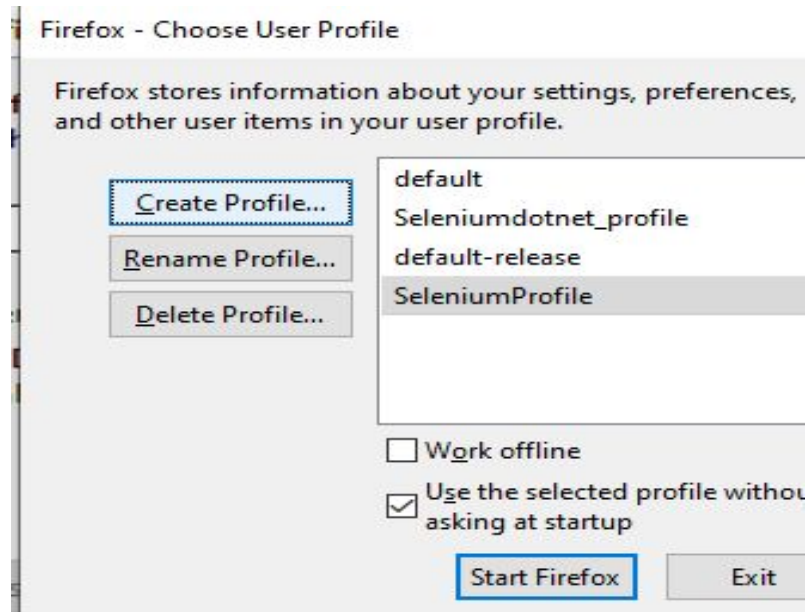


Figure 21

Now you can use this profile in your selenium scripts by writing below 2 lines

```
FirefoxProfileManager fpm = new FirefoxProfileManager();  
FirefoxProfile fp = fpm.GetProfile("SeleniumProfile");
```

You can read more here, see below

The screenshot shows the Selenium Java API documentation for the `FirefoxProfile` class. The browser address bar shows `selenium.dev/selenium/docs/api/java/index.html`. On the left, a sidebar lists 'All Classes' and 'Packages'. The 'Packages' list includes `com.thoughtworks.selenium` and `org.openqa.selenium`. The 'All Classes' list includes `FindFirstSelectedOption`, `FindsByClassName`, `FindsByCssSelector`, `FindsById`, `FindsByLinkText`, `FindsByName`, `FindsByTagName`, `FindsByXPath`, `FindSelectedOptionProc`, `FireEvent`, `FirefoxBinary`, and `FirefoxBinary.Channel`. The main content area displays the `Class FirefoxProfile` page. It shows the inheritance hierarchy: `java.lang.Object` and `org.openqa.selenium.firefox.FirefoxProfile`. The class declaration is `public class FirefoxProfile extends java.lang.Object`. Below this, a 'Field Summary' section is visible, containing a table with the following data:

Fields	
Modifier and Type	Field and
static java.lang.String	ALLOWED
static java.lang.String	PORT_PI

Figure 22

So in this tutorial, we learnt about how to handle java alert popups, notifications. We also learnt how to create and use firefox profile in our scripts. Thank you for reading!