Cookies testing using Selenium

# What is a cookie?

A HTTP cookie is information about the user and their preferences. It stores information using a key-value pair. It is a small piece of data sent from Web Application and stored in Web Browser, while the user is browsing that website.

# Selenium methods to support cookies testing

|  |  |
| --- | --- |
| Selenium Query Commands | Output |
| driver.manage().getCookies(); | Return The List of all Cookies |
| driver.manage().getCookieNamed(arg0); | Return specific cookie according to name |
| driver.manage().addCookie(arg0); | Create and add the cookie |
| driver.manage().deleteCookie(arg0); | Delete specific cookie |
| driver.manage().deleteCookieNamed(arg0); | Delete specific cookie according Name |
| driver.manage().deleteAllCookies(); | Delete all cookies |

When[Testing](https://www.guru99.com/software-testing.html)a web application using the selenium web driver, you may need to create, update, or delete a cookie.

For example, when automating an Online Shopping Application, you must automate test scenarios like placing an order, viewing a Cart, Payment Information, order confirmation, etc.

If cookies are not stored, you must perform login action every time before executing the above-listed test scenarios. This will increase your coding effort and execution time.

The solution is to store cookies in a File. Later, retrieve the cookie values from this file and add to it your current browser session. As a result, you can skip the login steps in every[Test Case](https://www.guru99.com/test-case.html).

## Storing cookie

**public** **void** storeCookie(WebDriver driver) {

File file = **new** File("D:\\Training\\Cookies\\Cookies.data");

**try**

{

// Delete old file if exists

file.delete();

file.createNewFile();

FileWriter fileWrite = **new** FileWriter(file);

BufferedWriter Bwrite = **new** BufferedWriter(fileWrite);

// loop for getting the cookie information

**for**(Cookie ck : driver.manage().getCookies())

{

Bwrite.write((ck.getName()+";"+ck.getValue()+";"+ck.getDomain()+";"+ck.getPath()+";"+ck.getExpiry()+";"+ck.isSecure()));

Bwrite.newLine();

}

Bwrite.close();

fileWrite.close();

}

**catch**(Exception ex)

{

ex.printStackTrace();

}

}

## Using stored cookie to login

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

driver.get("https://magento.softwaretestingboard.com/customer/account/login");

**try**{

File file = **new** File("D:\\Training\\Cookies\\Cookies.data");

FileReader fileReader = **new** FileReader(file);

BufferedReader Buffreader = **new** BufferedReader(fileReader);

String strline;

**while**((strline=Buffreader.readLine())!=**null**){

String[] s = strline.split(";");

String name = s[0];

String value = s[1];

String domain = s[2];

String path = s[3];

Date expiry = **null**;

**try** {

DateFormat format = **new** SimpleDateFormat("E MMM dd HH:mm:ss z yyyy", Locale.***ENGLISH***);

expiry = format.parse(s[4]);

}

**catch**(Exception ex) {

System.***out***.println("No date");

}

Boolean isSecure = Boolean.*parseBoolean*(s[5]);

Cookie ck = **new** Cookie(name,value,domain,path,expiry,isSecure);

driver.manage().addCookie(ck); // This will add the stored cookie to your current session

}

}**catch**(Exception ex){

ex.printStackTrace();

}

driver.get("https://magento.softwaretestingboard.com/customer/account/");

# Chrome options and desired capabilities

The **Chromeoptions Class** is a concept in Selenium WebDriver for manipulating various properties of the Chrome driver. It helps us perform various operations like opening Chrome in maximized mode, disabling existing extensions, disabling pop-ups, etc.

Ex: Using Chrome options to open browser in maximized form.

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

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

Arguments available for Chrome Options

* **start-maximized**: Opens Chrome in maximize mode
* **incognito:**Opens Chrome in incognito mode
* **headless:** Opens Chrome in headless mode
* **disable-extensions**: Disables existing extensions on Chrome browser
* **disable-popup-blocking**: Disables pop-ups displayed on Chrome browser
* **make-default-browser:** Makes Chrome default browser
* **version**: Prints chrome browser version
* **disable-infobars:** Prevents Chrome from displaying the notification ‘Chrome is being controlled by automated software

## Desired Capabilities

**Desired Capabilities Class** is used to modify multiple properties of web drivers. It provides key-value pairs to change individual properties of web drivers such as browser name, browser platform, etc. A common method of Desired Capabilities class is the setCapability method. It is mostly used with Selenium Grid, where the same test case needs to be executed on different browsers.