1. **How to handle SSL Certificate Error in selenium?**

**ANS :**

**What is SSL Certificate ?**

SSL (Secure Sockets Layer) is a standard security protocol for establishing a secure connection between the server and the client which is a browser.

There are number of benefits of using SSL certificate like,

* One can increase their users' and customer's trust in order to enhance the business' growth rapidly
* These certificates help to secure online transactions and customers sensitive information like credit-card/debit-card data, etc.
* Signing certificate tends to get a maximum number of downloads and good reviews from users.

SSL-secured websites begin with **https://**and you can see a lock icon or green address bar if the connection is securely established.

* When secure connection is not established between the server and client due to certificate SSL certificate error will occur

**Types of SSL Certificate Error**

Suppose you type some https request in the browser and get a message such as "This connection is Untrusted" or the "The site's security certificate is not trusted" depending upon the browser you are using. Then such error is subject to SSL certificate error.

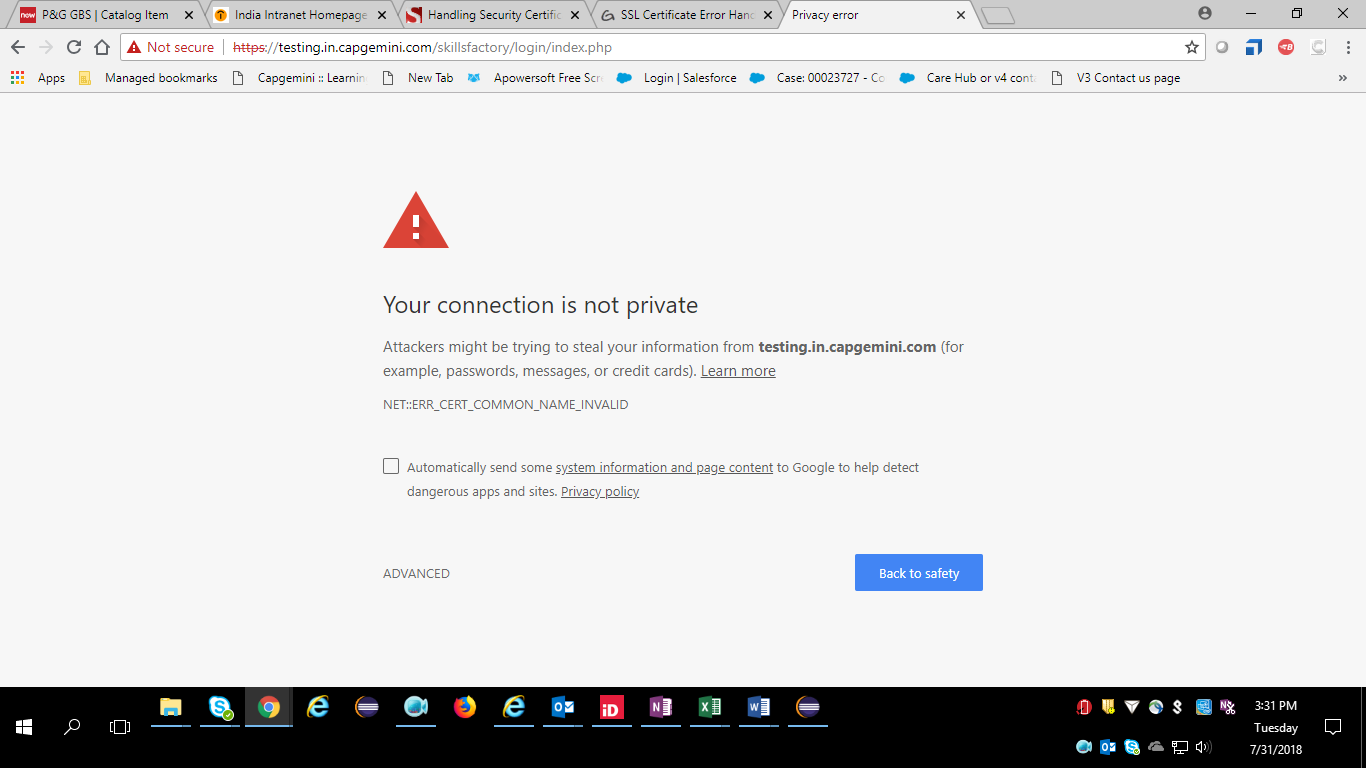
Now, if the browser is unable to establish a secured connection with the requested certificate, then the browser will throw "Untrusted Connection" exception as below and ask the user to take appropriate action.

The types of error you likely to see due to certificate in different browsers may be somewhat like this

1. **FireFox** - This connection is untrusted

# **Google Chrome** - Your connection is not private

1. **Internet Explorer ( IE)** - This security certificate presented by this website was not trusted by a trusted certificate authority (CA)



* Need to adjust our script in such a way that it will take care of SSL Exception/error by itself through Selenium Web driver.

## SSL Certificate Error Handling in Chrome

DesiredCapabilities handlSSLErr = DesiredCapabilities.chrome ()

handlSSLErr.setCapability (CapabilityType.ACCEPT\_SSL\_CERTS, true)

WebDriver driver = new ChromeDriver (handlSSLErr);

## SSL Certificate Error Handling in IE

DesiredCapabilities capabilities = new DesiredCapabilities();

capabilities.setCapability(CapabilityType.ACCEPT\_SSL\_CERTS, true);

System.setProperty("webdriver.ie.driver","IEDriverServer.exe");

WebDriver driver = new InternetExplorerDriver(capabilities);

I observed that in selenium 3.8.0

WebDriver driver3=new ~~InternetExplorerDriver~~(dc); 🡨 This is deprecated

So I can use it like

InternetExplorerOptions ieo=**new** InternetExplorerOptions();

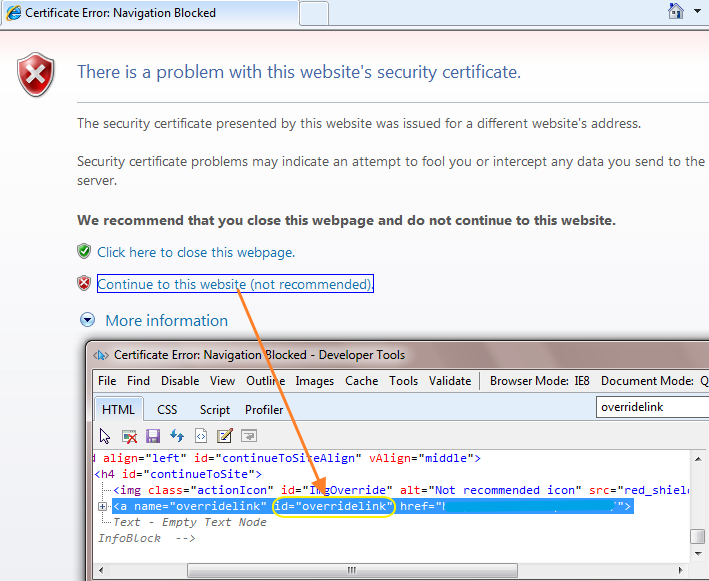
ieo.setCapability(CapabilityType.***ACCEPT\_SSL\_CERTS***, **true**);

WebDriver driver3=**new** InternetExplorerDriver(ieo);

In IE browser, we may have to handle it using javascript like like below :

**//To click on "Continue to this website (not recommended)." link to load original website.**

driver.navigate().to("**javascript:document.getElementById('overridelink').click()**");



## SSL Certificate Error Handling in Firefox

FirefoxProfile ffp=new FirefoxProfile();

ffp.setAcceptUntrustedCertificates(true);

ffp.setAssumeUntrustedCertificateIssuer(false);

FirefoxOptions ffo=new FirefoxOptions();

ffo.setProfile(ffp);

WebDriver driver1=new FirefoxDriver(ffo);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. **How to handle hidden elements ?**
   1. **How to click on hidden element**

**Mathod 1):** WebDriver **element** = webdriver.findElement(By.xpath(“<locator>”));

JavascriptExecutor **Executor** = (JavascriptExecutor)webdriver);

Executor.executeScript(“**arguments[0]**.click();”, **element**);

* 1. **How to get the Text from hidden elements?**

**Method 1 ):** In some cases, one may find it useful to get the hidden text, which can be retrieved from element's **textContent, innerText** or **innerHTML** attribute, by calling **element.getAttribute('attributeName')**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***