Page & Script Load Timeout in Selenium

Table of content

Timeouts Interface in Selenium Webdriver

Timeouts interface manages all the waits of the driver instances; This is the inner Interface of Webdriver Interface; in other words, the Timeouts interface is enclosed by the **Webdriver interface**.

Timeouts interface has three abstract methods, which are :

implicitlyWait

setScriptTimeout

pageLoadTimeout

There is no implementation present for these methods in the Timeouts interface; the browser classes(FirefoxDriver, ChromeDriver..) provides the implementations for these methods because the browser classes implement the Webdriver Interface.

Page Load Timeout in Selenium Webdriver

Page load timeout in selenium requests/set the time limit for a page to load, If the page is not loaded within the given time frame selenium throws **TimeOutException**.

We can set the page load timeout using the pageLoadTimeout method present in Browser classes(FirefoxDriver, ChromeDriver..)

```
driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);
```

Page load timeout is useful when we perform a performance test, or when we test execution in IE.

Page Load timeout is applicable only to `driver.get()` and `driver.navigate().to()` methods in **selenium**

Setting Negative time limit makes the selenium to wait for the page load infinitely

```
// set page load time as infinite (by giving minus value)
driver.manage().timeouts().pageLoadTimeout(-10, TimeUnit.SECONDS);
```

> Page load timeout is not applicable when the user clicks a link to open a page.

Complete program for Page load timeout

```
public class PageLoadTimeTest {
   public static void main(String[] args) {
      // set chrome driver exe path
      System.setProperty("webdriver.chrome.driver", "C:/~/chromedriver.exe");
```

```
        System.setProperty("webdriver.chrome.driver", "C:/Chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://chercher.tech/");
        // set the time of 30 seconds for page to complete the loading
        driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);
    }
}
```

Set Script Load timeout

setScriptTimeout sets the time limit for the asynchronous script to finish execution, if the process is not completed before the time limit, **selenium** throws **TimeOutException**

The setScriptTimeout method affects only **JavaScript code** executed with executeAsyncScript and nothing else. In particular, executeScript is not affected by it.

So why do you want to set a timeout for executeAsyncScript?

The default timeout for setScriptTimeout method is 0 (zero) if we do not set any time our executeAsyncScript method may fail because the **JavaScript code** may take more than zero seconds. So to avoid unnecessary failures, we have to set the setScriptTimeout.

Run a simple javascript: (Do not set setScriptTimeout) - Now this shall execute without throwing any issue.

```
((JavascriptExecutor) driver).executeScript("alert('I am alert');");
```

Run a simple Async Script: (Do not set setScriptTimeout) - This shall fail with error - "Timed out waiting for async script result after 0ms."

```
((JavascriptExecutor) driver).executeAsyncScript("document.getElementById('dummy')");
```

To resolve above issue add below-given setScriptTimeout

```
driver.manage().timeouts().setScriptTimeout(10, TimeUnit.SECONDS);
((JavascriptExecutor) driver).executeAsyncScript("document.getElementById('dummy')");
```

Recommended Readings

- **Implicit Wait in Selenium**
- **Fluent Wait in Selenium**
- **Explict Wait in Selenium**
- **Css Selector in Selenium**
- **Strings in Java & Selenium**
- **Sendkeys : Click : Clear : Submit ~ Element Input Operations**
- **Get Text, Attribute, CSS, Size values from Element in Selenium**
- **Find Element / Find Elements in Selenium**