

Get Accurate Timers in Web

Using Navigation Timing from Dom on real devices

by Uzi Eilon

Access to DOM allows the client to build robust automation by finding the web elements and execute actions on these elements , but the DOM contains more data. In this article I will show how to get the timers from the DOM and use it as part of Selenium script.

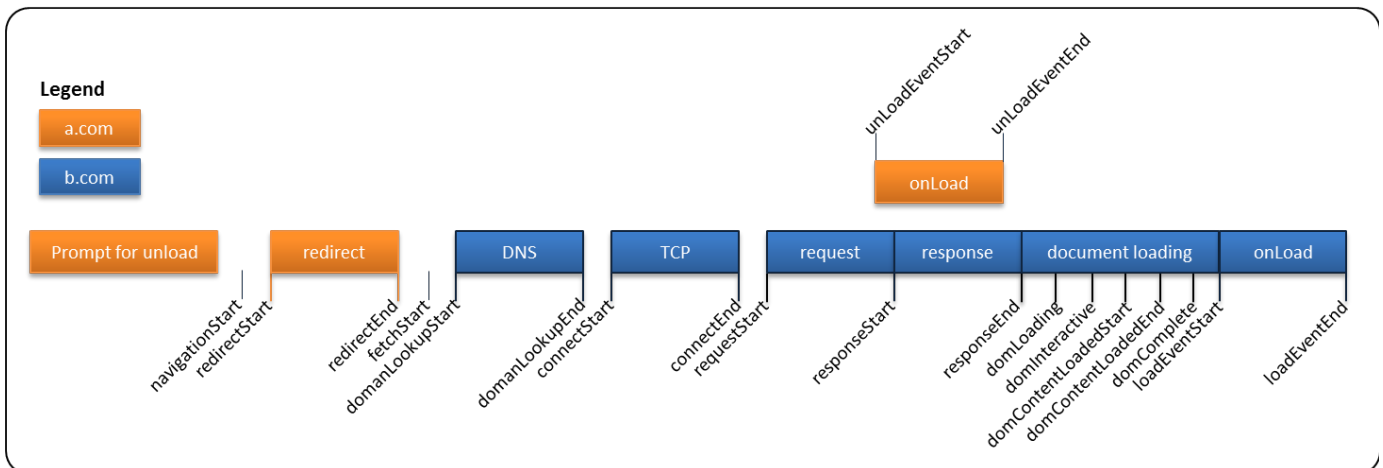
How to get the timers from the DOM

After the page is loaded you can access to the document/window and fetch data, I'm getting the Navigate Timing using command: **window.performance.timing**

The results is array with the following data:

```
connectEnd = 1442838557626
connectStart = 1442838557526
domComplete = 1442838557915
domContentLoadedEventEnd = 1442838557753
domContentLoadedEventStart = 1442838557753
domInteractive = 1442838557753
domLoading = 1442838557691
domainLookupEnd = 1442838557526
domainLookupStart = 1442838557526
fetchStart = 1442838557523
loadEventEnd = 1442838557915
loadEventStart = 1442838557915
navigationStart = 1442838557300
redirectEnd = 0
redirectStart = 0
requestStart = 1442838557626
responseEnd = 1442838557752
responseStart = 1442838557657
secureConnectionStart = 1442838557554
unloadEventEnd = 0
unloadEventStart = 0
```

The following figure explains the different timers :



The following code gets the data from RemoteWebDriver and generate timers report.
I used Perfecto Mobile RemoteWebDriver and get the timers from real devices.

```
import java.util.HashMap;
import java.util.Map;
import org.openqa.selenium.remote.RemoteWebDriver;
public class NavigateTimeReport {

public static void printPageTimers(RemoteWebDriver w)

    {

        Map<String,String> params = new HashMap<String,String>();

        Object size = w.executeScript("var a = window.performance.timing
; return a; ", params);

        printTimers ((Map<String, Long>) size);

    }

private static void printTimers( Map<String, Long> data)
{
    long navStart = data.get("navigationStart");
    long loadEventEnd = data.get("loadEventEnd");
    long connectEnd = data.get("connectEnd");
    long requestStart = data.get("requestStart");
    long responseStart = data.get("responseStart");
    long responseEnd = data.get("responseEnd");
    long domLoaded = data.get("domContentLoadedEventStart");

    System.out.println(" ** PAGE TIMERS **");
    System.out.println("Total Duration := "+(loadEventEnd - navStart)
);
    System.out.println("Network (redirect ,DNS,TCP) := "+(connectEnd -
navStart) );
    System.out.println("httpRequest := "+(responseStart -
requestStart) );
    System.out.println("httpRspnse := "+(responseEnd - responseStart)
);
    System.out.println("Build DOM := "+(domLoaded - responseEnd) );
    System.out.println("Rendring := "+(loadEventEnd - domLoaded) );
    System.out.println(" ** **** **");

}

}
```

The navigation timers can help us to analyzed performance issues and drill down into the real problem.
In order to use it start the webdriver , use get function , validate the page loaded by finding and element and then execute the getTimer code.