SELENIUM 4.0 NEW FEATURES AND UPDATES

BY ENGINEER DIARIES

PART-2

1.Chrome Dev Protocol

2. Geo location Feature

3. Device Configuration Feature

4. Capturing Console logs

5. Capturing Performance metrics

6. Network Interception setup

Engineer Diaries

Introduction to Chrome Dev Protocol

1. Selenium webdriver works on request→response mechanism. That is whatever request you send to webdriver it sends using w3 standard protocol to respective browser and in webdriver what we sees is in fact a response of the API.
2. These webdrivers are also provided by the browser vendors themselves and selenium mere uses them.
3. The developer tool window which we have full of elements, console, network tab and performance of the site.
4. The developer tool window in fact in many case scenarios is used for API and performance testing. So, with this you are now being able to use Selenium for much more than just UI/UX(User Interface/User Experience).

Below will have a look at what all you can do using chrome dev protocol.

Engineer Diaries

GEO LOCATION FEATURE

Now you can set user location using Chrome Dev Protocol which would help to test User experience in different Time Zones. Below we are sending coordinates of Delhi, India.

```
ChromeDriver driver = new ChromeDriver();
DevTools devTools = driver.getDevTools();
devTools.createSession();
devTools.send(Emulation.setGeolocationOverride(
Optional.of(28.7041),
Optional.of(77.1025),
Optional.of(1)));
driver.get("https://my-location.org/");
```

Engineer Diaries

Setting Device Configuration

Another interesting feature in chrome dev tool protocol is you can set device screen size. This feature helps to test responsiveness of the site across various screen size devices. The web developers make their site responsive so that the user irrespective of which screen size he is using is still able to get best possible user experience.

```
DevTools devTools = driver.getDevTools();
devTools.createSession();
// iPhone 11 Pro dimensions
devTools.send(Emulation.setDeviceMetricsOverride(375,
                                                 812,
                                                 50,
                                                 true,
                                                 Optional.empty(),
                                                 Optional.empty(),
                                                 Optional.empty(),
                                                 Optional.empty(),
                                                 Optional.empty(),
                                                 Optional.empty(),
                                                 Optional.empty(),
                                                 Optional.empty(),
                                                 Optional.empty()));
driver.get("https://www.geeksforgeeks.org/");
```

Capturing console.logs

Console logs are the most important testing way to test API testing. The data you are getting from backend. With this feature now you can capture console.logs which means if you need to bulk test a site if the developer has removed these logs post production or check how many pages are getting affected by some server bug. It has opened selenium outlet into API testing world.

```
DevTools devTools = driver.getDevTools();
devTools.createSession();
devTools.send(Log.enable());
devTools.addListener(Log.entryAdded(),
logEntry -> {
System.out.println("log: "+logEntry.getText());
System.out.println("level: "+logEntry.getLevel());
                    });
driver.get("https://www.geeksforgeeks.org/");
```

Engineer Diaries

Capturing performance metrics

This performance metric feature makes Selenium tool a performance tool as well. You can put criteria of speed test, and page loading capacity of a site. The following code would print all the performance metrics.

```
DevTools devTools = driver.getDevTools();
devTools.createSession();
devTools.send(Performance.enable(Optional.empty()));
List<Metric> metricList =
devTools.send(Performance.getMetrics());
driver.get("https://google.com");
for(Metric m : metricList)
{
System.out.println(m.getName() + " = " + m.getValue())
}
```

Network Interception using Dev Tool

Till now if you had to check response of an API or webpage you had to use java.net along with Selenium. This was the most common way to check if a link is working or not. But now selenium has provided Network class in Dev tool where you can send in request headers and check in the response as well. In below code we are sending additional headers to a webpage.

```
DevTools devTools = driver.getDevTools();
devTools.createSession();
devTools.send(Network.enable(Optional.empty(),
Optional.empty(), Optional.empty()));
Headers headers = new
Headers(Collections.singletonMap("hello", "world"));
devTools.send(Network.setExtraHTTPHeaders(headers));
driver.get("https://manytools.org/http-html-text/http-
request-headers/");
```

Engineer Diaries