

Use Timers in Automation Testing (Appium/Selenium)

by Christopher Alton and Amin Caceres

Perfecto supports different methods when using timers.

Eclipse/IntelliJ (JAVA)

This JAVA method will allow for the use of any generic timer. In this sample, we are using stopwatch as the timer dependency. We then pair this with the [Timer Report function](#) to put the stopwatch timer into my execution report.

Sample:

```
//Create new stopwatch timer
StopWatch stopwatch = new StopWatch();
stopwatch.start();

// Go to some website so just this script does something
driver.get("https://www.nfl.com");

// Here I would wait for some element to show up but I'll just use a hard wait
Thread.sleep(10000);

// Stop the StopWatch
stopwatch.stop();

//Get the timer and put it in a variable
long x = stopwatch.getTime();

//Convert the result to a string
String numberAsString = Long.toString(x);

//Report it to Perfecto
Map<String, Object> params1 = new HashMap<>();
params1.put("name", "stopwatch timer");
params1.put("result", numberAsString);
Object result1 = driver.executeScript("mobile:status:timer", params1);
```

This sample here, is a fully working method with dependencies. Fill in the user name, password, cloud name (host) and the device ID. This test should run on any mobile Android or iOS version.

Sample:

[TimerGenericTest.txt](#)

Please note that the results in the report, by default, are in milliseconds. If you wish to convert this time to seconds in your report, use a JAVA method to do so.

Native IDE (Perfecto Automation View)

When you are working within the Native IDE, which is the Perfecto Automation View within the Perfecto Lab via your web browser, you can use the built in timer functions. Here is a listing of those functions.

[Timer Start](#)

[Timer Stop](#)

[Get Timer](#)

Other timer methods can be integrated. It is up to you, the test writer, to find which method works the best for your needs.

Last updated: December 28th, 2018