* Set up guide for perfecto mobile automation using selenium in eclipse:
* JDK 1.8\_101

https://www.perfecto.io/platform/debug-with-devtunnel/selenium-in-the-cloud/

* Selenium Jars + Mobile jars

1. ashot-1.4.12
2. byte-buddy-1.7.5
3. client-combined-3.6.0
4. client-combined-3.6.0-sources
5. commons-codec-1.10
6. commons-collections4-4.1
7. commons-exec-1.3
8. commons-io-2.5
9. commons-lang3-3.6
10. commons-logging-1.2
11. curvesapi-1.04
12. dom4j-1.6.1
13. excel-1.3.4-sources
14. excelutils-1.41
15. gson-2.8.0
16. guava-23.0
17. httpclient-4.5.3
18. http-client-10.9.0.1
19. httpcore-4.4.6
20. jackson-annotations-2.8.5
21. jackson-core-2.8.5
22. jackson-databind-2.8.5
23. jasmine-1.3.1
24. java-client-3.4.1
25. java-client-5.0.4
26. jaxen-1.1.6
27. jcl-over-slf4j-1.7.22
28. jna-4.1.0
29. jna-platform-4.1.0
30. json-20160810
31. junit-4.12
32. jxl-1.0
33. log4j-1.2.17
34. logback-classic-1.1.8
35. logback-core-1.1.8
36. org.mortbay.jetty
37. pm-webdriver-10.9.0.1
38. poi-3.17
39. poi-examples-3.17
40. poi-excelant-3.17
41. poi-ooxml-3.17
42. poi-ooxml-schemas-3.17
43. poi-scratchpad-3.17
44. reportium-java-1.1.9
45. selenium-api-2.53.0
46. selenium-remote-driver-2.53.0
47. selenium-server-standalone-3.6.0
48. selenium-server-standalone-3.7.1
49. selenium-server-standalone-3.13.0
50. selenium-support-2.53.0
51. slf4j-api-1.7.22
52. xmlbeans-2.6.0

**DriverClass (Contain Main method)**

package MobPackage;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.Platform;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

import java.io.IOException;

import java.net.MalformedURLException;

import java.net.URL;

import org.openqa.selenium.Platform;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

public class driverClass {

public static void main(String[] args) throws InterruptedException, MalformedURLException {

ConfigFileReader configFileReader= new ConfigFileReader();

//This code will run the test on Perfecto Mobile Cloud

String browserName ="mobileOS";

DesiredCapabilities capabilities = new DesiredCapabilities(browserName, "", Platform.ANY);

// Enter your cloud URL

String host = "bmo-qaed.perfectomobile.com";

capabilities.setCapability("host", configFileReader.getRemotePath());

capabilities.setCapability("user", configFileReader.getpfuser());

capabilities.setCapability("password",configFileReader.getpfpassword());

// capabilities.setCapability("mobile:browser:clean", configFileReader.pfdevice());

capabilities.setCapability("mobile:browser:goto", configFileReader.pfdevice());

capabilities.setCapability("mobile:browser:open", configFileReader.pfdevice());

// capabilities.setCapability("password","Perfecto2017");

// capabilities.setCapability("deviceName", "075B61861CC330A5D2FB4957CD8376AB15DA55DD");

capabilities.setCapability("deviceName", configFileReader.pfdevice());

// capabilities.setCapability("rotatable", true);

//Proxy

System.getProperties().put("http.proxyHost",configFileReader.getproxyhost());

System.getProperties().put("http.proxyPort", "8080");

System.getProperties().put("https.proxyHost", configFileReader.getproxyhost());

System.getProperties().put("https.proxyPort", "8080");

RemoteWebDriver driver = new RemoteWebDriver(new URL("https://" + host + "/nexperience/perfectomobile/wd/hub"), capabilities);

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

try {

//Launch the chrome browser and navigate to www.google.com

driver.get("http://www.google.com");

// Print a Log In message to the screen

System.out.println("Successfully opened the website www.google.com");

//Wait for 3 Sec

Thread.sleep(3000);

//Search Perfecto Mobile

driver.findElement(By.xpath("//\*[@class='gLFyf']")).sendKeys("Perfecto Mobile\n");

// Click on PErfecto Mobile and navigate to the web-site.

driver.findElement(By.linkText("Perfecto Mobile")).click();

//Wait for 15 Sec

Thread.sleep(15000);

}

finally {

// Close the driver

driver.quit();

}

}

}

**ConfigFileReader : To read property value to set capabilities:**

package MobPackage;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.util.Properties;

public class ConfigFileReader {

private Properties properties;

private final String propertyFilePath= "C://Users//rdas06//Desktop//Selenium//MobileDemo//src//MobPackage//Configuration.properties";

public ConfigFileReader(){

BufferedReader reader;

try {

reader = new BufferedReader(new FileReader(propertyFilePath));

properties = new Properties();

try {

properties.load(reader);

reader.close();

} catch (IOException e) {

e.printStackTrace();

}

} catch (FileNotFoundException e) {

e.printStackTrace();

throw new RuntimeException("Configuration.properties not found at " + propertyFilePath);

}

}

public String getpfuser(){

String driverPath = properties.getProperty("pfuser");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("User details not specified in the Configuration.properties file.");

}

public String getpfuser1(){

String driverPath = properties.getProperty("pfuser1");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("User details not specified in the Configuration.properties file.");

}

public String getpfpassword(){

String driverPath = properties.getProperty("pfpassword");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("Password not specified in the Configuration.properties file.");

}

public String getpfplatform(){

String driverPath = properties.getProperty("pfplatform");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("Platform not specified in the Configuration.properties file.");

}

public String getpfplatformversion(){

String driverPath = properties.getProperty("pfplatformversion");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("platformversion not specified in the Configuration.properties file.");

}

public String getpfbrowser(){

String driverPath = properties.getProperty("pfbrowser");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("Pfbrowser host not specified in the Configuration.properties file.");

}

public String getproxyhost(){

String driverPath = properties.getProperty("proxyhost");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("proxyhost not specified in the Configuration.properties file.");

}

public String getpfbrowserversion(){

String driverPath = properties.getProperty("pfbrowserversion");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("Browser version not specified in the Configuration.properties file.");

}

public String getpfresolution(){

String driverPath = properties.getProperty("pfresolution");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("Resolution not specified in the Configuration.properties file.");

}

public String getpflocation(){

String driverPath = properties.getProperty("pflocation");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("Pflocation not specified in the Configuration.properties file.");

}

public String pfdevice(){

String driverPath = properties.getProperty("pfdevice");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("Device details not specified in the Configuration.properties file.");

}

public String getRemotePath(){

String driverPath = properties.getProperty("remotehost");

if(driverPath!= null) return driverPath;

else throw new RuntimeException("remote host not specified in the Configuration.properties file.");

}

public long getImplicitlyWait() {

String implicitlyWait = properties.getProperty("implicitlyWait");

if(implicitlyWait != null) return Long.parseLong(implicitlyWait);

else throw new RuntimeException("implicitlyWait not specified in the Configuration.properties file.");

}

public String getApplicationUrl() {

String url = properties.getProperty("url");

if(url != null) return url;

else throw new RuntimeException("url not specified in the Configuration.properties file.");

}

}