Setting up appium

Team 12, Project mobile development

Inhoud

[Step 1: Download and install prerequisites 2](#_Toc419888459)

[Step 2: Setting up environment variables 3](#_Toc419888460)

[Step 3: Enabling USB Debugging on your mobile 4](#_Toc419888461)

[Step 4: Creating a project and adding the libraries 5](#_Toc419888462)

[Step 5: Starting appium 7](#_Toc419888463)

# Step 1: Download and install prerequisites

Before we start you have to download and install the following programs:

1. Java Development Kit

You can download this from: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

1. Android studio

You can download this from: <https://developer.android.com/sdk/index.html>

1. Node.js

You can download this from: https://nodejs.org/download/

1. Microsoft .net Framework

You can download this from: https://www.microsoft.com/nl-nl/download/details.aspx?id=17851

1. Selenium

You can download this from: http://www.seleniumhq.org/download/

1. Appium java client jar

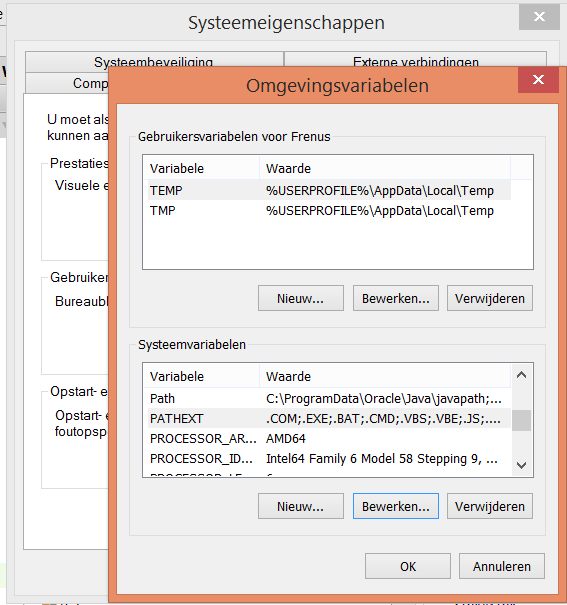
You can download this JAR from: https://search.maven.org/#search%7Cga%7C1%7Cg%3Aio.appium%20a%3Ajava-client

1. Appium

You can download this from: <http://appium.io/>

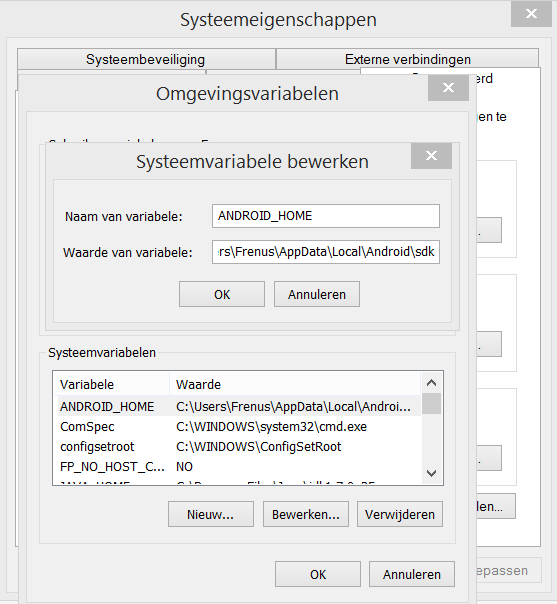
# Step 2: Setting up environment variables

Now that we have installed these programs we need to set them in our system environment variables so appium can locate them. To do this go to My computer properties -> Advanced system settings -> Environment variables. You now have the following screen:



To add our java jdk and android sdk we have to make 2 new System variables.

Click the New button and add the paths to your android and java folder. Call them ANDROID\_HOME and JAVA\_HOME like this:



My paths were:

ANDROID\_HOME C:\Users\Frenus\AppData\Local\Android\sdk, and

JAVA\_HOME C:\Program Files\Java\jdk1.7.0\_25

The last step is to add these to our already existing Path system variable.

Click edit and add the following lines at the end:

;%JAVA\_HOME%\bin;%ANDROID\_HOME%\tools;%ANDROID\_HOME%\platform\_tools

# Step 3: Enabling USB Debugging on your mobile

We need to enable USB Dubugging on the device you use so appium can access your mobile. To enable this you first have to enable developer Options. While enabling Developer Options is done in the same way for every Android phone or tablet, OEMs don’t always put the option in the same place. Navigate your phone to the “Build number” portion of the settings, which can be tucked away and buried in submenus.

Here’s how to get there on a few popular devices:

Stock Android: Settings > About phone > Build number

Samsung Galaxy S5: Settings > About device > Build number

LG G3: Settings > About phone > Software information > Build number

HTC One (M8): Settings > About > Software information > More > Build number

Once you’ve found the Build number section of the settings, tap on the section 7 times. After two taps, a small pop up notification should appear saying "you are now X steps away from being a developer" with a number that counts down with every additional tap.

After the 7th tap, the Developer options will be unlocked and available. Click it and enable USB debugging.

# Step 4: Creating a project and adding the libraries

For some sample code I downloaded a project from the saucelabs github: https://github.com/appium/sample-code/tree/master/sample-code/examples/java/junit/src/test/java/com/saucelabs/appium

To get this to run we need to add the .jar files

G to File -> Project Structure -> module:app -> Dependencies.

Press the + too add the following jars: All Selenium jars and the Java client jar.

There will still be some errors because there are some jars missing, but you can right click them and download them easily.

We have to change the code a bit to get this working, because we use another project. Here is the code I used for our application:

package com.saucelabs.appium;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.remote.DesiredCapabilities;

import java.net.URL;

import io.appium.java\_client.AppiumDriver;

import io.appium.java\_client.android.AndroidDriver;

import static org.junit.Assert.assertEquals;

public class AndroidTest {

private AppiumDriver driver;

@Before

public void setUp() throws Exception {

DesiredCapabilities capabilities = new DesiredCapabilities();

capabilities.setCapability("deviceName", "Nexus 5");

capabilities.setCapability("platformVersion", "4.4");

capabilities.setCapability("appPackage", "com.testapplication.wfcmainpage");

capabilities.setCapability("activity.MainActivity", "appActivity");

driver = new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"), capabilities);

}

@After

public void tearDown() throws Exception {

driver.quit();

}

@Test

public void testButtons() throws InterruptedException {

driver.findElement(By.id("rentButton")).click();

driver.navigate().back();

driver.findElement(By.id("facilitiesButton")).click();

driver.navigate().back();

driver.findElement(By.id("infoButton")).click();

driver.navigate().back();

driver.findElement(By.id("navigationButton")).click();

driver.navigate().back();

}

@Test

public void testFacilities() throws InterruptedException {

driver.findElement(By.id("facilitiesButton")).click();

driver.findElements(By.id("customRowText")).get(0).click();

assertEquals("2CLASHY", driver.findElement(By.id("txtFacilityName")).getText());

}

}

# Step 5: Starting appium

To start the appium server you first have to run the .exe, then you have to press the play button in the right upper corner. Once you see the >info: Console LogLevel: debug you can run your tests

