**ERROR LOGS & Their Resolutions**

* The connection to adb is down, and a severe error has occured.

**Resolution:**

1. Close the Eclipse if running
2. Go to the Android SDK platform-tools directory in Command Prompt
3. type adb kill-server (**Eclipse should be closed before issuing these commands**)
4. then type adb start-server
5. No error message is thrown while starting ADB server, then adb is started successfully.
6. Now you can start Eclipse again.

In CMD while killing or starting adb server, you may get error as “Unable to connect to daemon etc etc”

1. Use task manager to kill all process related adb (adb.exe32)
2. Close ecipse
3. Restart the system
4. Open cmd
5. Navigate to android sdk->platform-tools folder
6. Adb start-server

* [ADB Android Device Unauthorized](http://stackoverflow.com/questions/23081263/adb-android-device-unauthorized)

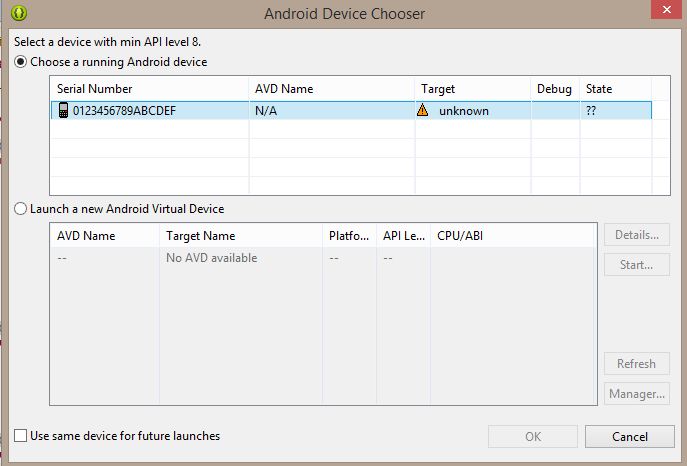
In cmd while executing adb device we get below command

<ANDROID\_SDK\_HOME>\platform-tools>adb devices

List of devices attached

4df798d76f98cf6d unauthorized

Or while executing the project in Eclipse we get the below screenshot:



**Resolution**

It's likely that the device is no longer authorized on [ADB](http://developer.android.com/tools/help/adb.html) for whatever reason.

**1. Check if authorized:**

<ANDROID\_SDK\_HOME>\platform-tools>adb devices

List of devices attached

4df798d76f98cf6d unauthorized

**2. Revoke USB Debugging on phone**

If the device is shown as **unauthorized**, go to the developer options on the phone and click **"Revoke USB debugging authorization"** (tested with JellyBean & Samsung GalaxyIII).

**3. Restart ADB Server:**

Then restarted adb server

adb kill-server

adb start-server

**4. Reconnect the device**

The device will ask if you are agree to connect the computer id. You need to confirm it.

**5. Now Check the device**

It is now authorized!

adb devices

<ANDROID\_SDK\_HOME>\platform-tools>adb devices

List of devices attached

4df798d76f98cf6d device

* **While writing code for native App:**

**public** **static** AppiumDriver *driver*;

DesiredCapabilities capabilities = **new** DesiredCapabilities();

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

In the last line, getting error as “Can not instantiate the type as Appium Server”

****

**Resolution:**

**Java –client 2.0 onwards, AppiumDriver has become AndroidDriver for ANDROID & IOSDriver for MAC/i**OS. E.g:

**public** **static** AndroidDriver *driver*;

DesiredCapabilities capabilities = **new** DesiredCapabilities();

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

# [NoClassDefFoundError in Java: com/google/common/base/Function](http://stackoverflow.com/questions/5134953/noclassdeffounderror-in-java-com-google-common-base-function)

**Resolution:**

I had the same problem, and finally I found that I forgot to add the [selenium-server-standalone-version.jar](https://code.google.com/p/selenium/downloads/list). I had only added the client jar, [selenium-java-version.jar](https://code.google.com/p/selenium/downloads/list).

Hope this helps.

# Type Mismatch:[Cannot convert from WebElement to List<WebElement>](http://stackoverflow.com/questions/29606900/cannot-convert-from-webelement-to-listwebelement)

You should use findElements to find the list of WebElements. See API doc [here](http://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/WebDriver.html#findElements-org.openqa.selenium.By-)

findElement returns single WebElement whereas findElements is plural and should be the expected one in this case.

List<WebElement> fields = driver.findElements(By.xpath("//input[@type='text']"));

System.out.println(fields.size());

# A new session could not be created.

Original error: An unknown server-side error occurred while processing the command. (Original error: unknown error: Chrome version must be >= 43.0.2357.0

# Remote Debugging on Android with Chrome

<https://developer.chrome.com/devtools/docs/remote-debugging>

