Fixing Bluetooth debugging on Android Wear

20 October 2014

Tags: groovy (/blog/tags/groovy.html) android (/blog/tags/android.html) moto 360 (/blog/tags/moto-360.html) wear (/blog/tags/wear.html)

I have been working on making Groovy (http://beta.groovy-lang.org) work on Android for several months now. In the last weeks, I even showed at SpringOne2GX an example of an application written in Groovy that worked on Android Wear. However, that code worked in an emulator. Recently, I got a real device, a Moto 360, so I wanted to see that application running on a real device. For some very obscure (understand buggy Android SDK) reason, it was far from being that easy...

I litteraly spent **hours** trying to figure out what was wrong, so I thought it would be interesting for those of you who face the same problem to have a blog post that explains how to deal with it.

The problem

If you follow the instructions on the Android documentation (https://developer.android.com/training/wearables/apps/bt-debugging.html) about how to enable bluetooth debugging, it's in the end pretty simple. Basically, it's about enabling USB debugging on your physical handheld, then enable bluetooth debugging on your wearable, and in the end enable bluetooth debugging in the Android Wear companion app.

The guide says:

> In the Android Wear companion app, you should see the status change to:
> Host: connected Target: connected

However, whatever I did, the `Target: connected` line never appeared for me. It was **always** `Target: disconnected`, so if I continued with the instructions:

```
adb forward tcp:4444 localabstract:/adb-hub; adb connect localhost:4444
```

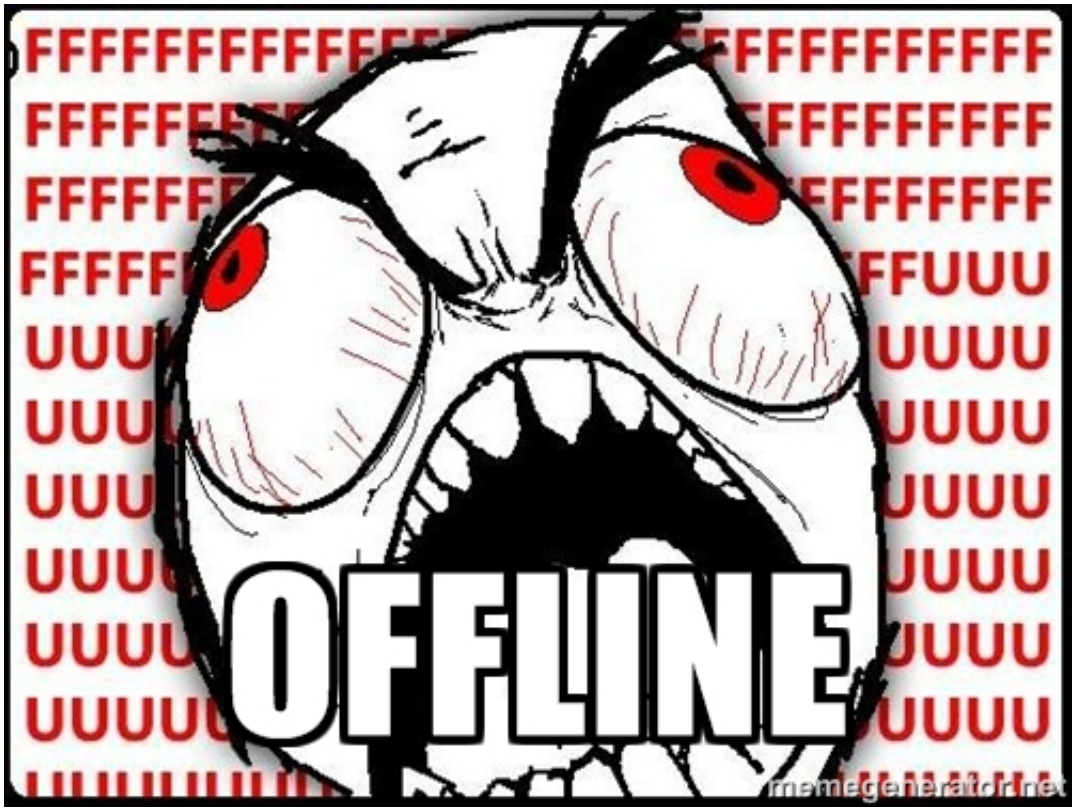
Doing

```
adb devices
```

Only showed my wearable as `offline`:

```
List of devices attached

dcfbbafd        device
localhost:4444  offline
```



I have searched for an answer in a lot of pages, including Stackoverflow (http://stackoverflow.com/questions/25938998/debugging-on-moto-360) where I tried very unlikely answers like in this page (http://stackoverflow.com/questions/25938998/debugging-on-moto-360): execute both commands separately instead of doing them in the same line. I thought that maybe there was a timing issue and that the fact of separating both commands would give the toolkit a chance, but no, wasn't that easy.

In the end, I was totally convinced that the problem was because I had previously associated my handheld with an emulator. I was convinced of it because even if I had now associated it with a real device (the Moto 360), in the Android Wear companion app, the device was recognized as an

"emulator"... mmm... So I tried uninstalling the Android Wear app, clear its cache, but no matter what I did, after reinstalling, the settings were kept, and the Moto recognized as an emulator. So sad...

The solution

So you were looking at a solution, and here it is. Basically, the problem is that the Android companion app doesn't store its settings under its cache. They are stored in the `Google Play Services` space, so here is the procedure that worked for me, and I sincerely hope it will do for you. On your handheld:

1. open the applications settings, search for the `Android Wear` application, then **force stop it**
2. clear its **data and cache**
3. now search for `Google Play Services` (depending on your language settings, it can appear with a different name, on m y device it is "Services Google Play".
4. click on `Manage space`. You will see that there's a section for connected devices. I tried to clear data here, but it didn't help, so you have to click on **delete all data**.
5. reboot your phone
6. reboot your Moto 360

I am unsure that the two last steps are really necessary, but I did it because I wanted to make sure that force stopping and clearing data did not introduce some weird behavior after that. When you reopen the Android Wear companion app, it should now be as if it was the first time you opened it and ask you to associate it with your watch. Do it, and now, you should be able to follow the normal procedure described in the Android documentation and...

Success, now you can debug your application on a real device (which includes deploying it...).

Cédric Champeau
@CedricChampeau

Suivre

Hey, finally got it working! I need to blog about how I arrived the insane emulator issue... #groovylang

13:30 - 19 Oct 2014

12 17

20 Comments My Blog 1 Login

Recommend 2 Share Sort by Best

Join the discussion…

ATT · 5 months ago

if you have connected with an emulator previously, then it will interfere with the connection. If that is the case, you don't need to do the whole reset thing.
All i had to do was to remove the emulator from the phone:
1. open Android wear companion app
2. Go to Settings
3. Click on Emulator
4. Click 'FORGET WATCH'

Worked for me

1 · Reply · Share ›

Naheel Azawy ↗ ATT · 2 months ago

THANK YOU!

Reply · Share ›

Tony Wickham ↗ ATT · 3 months ago

Make sure you are forgetting the emulator - not your real watch - because you will have to factory reset your watch to repair it to your phone.

Reply · Share ›

Jwalant Bhatt ↗ ATT · 4 months ago

worked for me too thank you !

Reply · Share ›

John Gu ↗ ATT · 4 months ago

This worked for me, didn't have to clear the data.

Reply · Share ›

Ganesh Krishnan · a month ago

For me it was as simple as unchecking and checking the "Debug over bluetooth" checkbox!

Reply · Share ›

T J · 4 months ago

TJ · 4 months ago

This is so fucking tedious.
My Moto 360 has stopped working with ADB. All I get is localhost:4444 unauthorized

and not even any confirmation dialog for debugging on my wear like I use to. How do I fix this stupid bug??

Reply · Share ›

Ganesh Krishnan ↗ TJ · a month ago

If you are on *nix systems (Ubuntu or Mac) run adb with sudo permissions. So sudo ./adb etc etc.

If you are on windows then you can just revoke the debugging permissions for all devices (Wear -> settings) and then try to connect again. Click accept on your computer to accept the Moto 360

Reply · Share ›

Aleksandr · 4 months ago

I think all the steps here and it didn't work for me when I was messing around with several phones and several watches. I ended up with one watch paired to one phone and adb thinking that that watch is still some old watch. Reconnecting to it didn't help. But what did help is to execute adb forward and adb connect commands with port 4445 instead of 4444.

Reply · Share ›

Aleksandr · 4 months ago

I'm having the same issue with two watches registered for a single Android Wear. No mater what I did to the re-enable debugging, it was always trying to reach out to my first watch and when it was disabled, it was shown as offline. It is funny that debugging option is global for Android Wear, it doesn't specify which watch you want to debug.

Reply · Share ›

Rameshd33 · 5 months ago

best one

Reply · Share ›

云峰 张 · 5 months ago

Worked for me too, thanks.

Reply · Share ›

Trish Whetzel · 6 months ago

Worked for me, rebooting both phone and watch was needed.

Reply · Share ›

Chris · 8 months ago

I found another fix for the same problem. Deleting my emulator virtual device causes adb connect localhost:4444 to connect through the phone, which pops up the rsa security key

confirmation and everything is good. So seems like the presence of an emulator confuses things.

Reply • Share ›

Robert E. Guinness • 9 months ago

Didn't work for me unfortunately.

Reply • Share ›

supergrover1337 • 9 months ago

This worked great! You do need to restart your phone to get it to connect.

Reply • Share ›

huteri • 10 months ago

You should make sure developer optionts is on on wear,

Reply • Share ›

3R1KU54J • 10 months ago

Thank you! Helped getting around "OFFLINE" on Nexus 7!

Reply • Share ›

Rubén López • 10 months ago

Thanks, if helped with my Samsung S4 + Sony Smartwatch3!!

Reply • Share ›

JC Del Valle • 9 months ago

Mmm... the offline status for the "localhost:4444" device is normal and correct, please see my quickly guide about it:
https://medium.com/@jcdelvalle...
Regards.

Reply • Share ›

```
static <t,e> List<t> collect(List<e> self, …
```
