**ANDROID STUDIO**



**CONNECT YOUR Devices to the PC/LAPTOP via USB**

**1. Make sure Developers Options MENU is enabled**

**( if not go to SettingsAbout Device Tap BUILD # 7 times**

**Go back one screen, find Developers Options Menu and go to # 2**

**2. Make sure USB DEBUGING BOX is CHECKED**

**3. “TRUST THIS COMPUTER” notification appears -CONFIRM YES**

**Open Android Studio**

**Connect your device, open Emulator**

**make sure United APP or LingoCard APP installed .**

**Emulator should have United APP installed.**

**adb devices**

Session 7

Another valuable command line to help developers debug (example shown from a real device):

clean up buffer : adb –d logcat -c

adb –d logcat –v threadtime > C:/a/class1.log

* Perform any actions on your Android device.
* To stop the adb process, press Ctrl + C on the command prompt window.

**Collecting threadtime logs from an app ( example Real Device)**

**adb –d logcat –c ( clean up buffer) before executing the following command.**

**adb –d logcat –v threadtime >C:\a\debug4.log | findstr com.united.mobile.android**

**or**

**adb –d logcat –v threadtime >debug4.log | adb –d shell grep com.united.mobile.android**

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Monkey Test is a TOOL in ANDROID Studio to perform stress test for UI functionality

5000 is a number of "hits" or in other words, “injections” your app is going to receive. You may change this number to any.

adb -d shell monkey -p com.united.mobile.android 5000

or

adb -d shell monkey -p com.lingocard.lingocard 5000

Collecting logs simulateneously you need to open another instance of Terminal , so both Android Studio terminal and Shell is open

cmd in Windows

Terminal in Mac OS If you want to collect logs from Monkey test, run shell **simultaneously** with this command line

adb -d logcat \*E >C:\a\monkey1.txt | findstr com.united.mobile.android

or

adb -d logcat \*E >monkey.txt | adb -d shell grep “com.united.mobile.android” ( from cmd shell )

Reboot your device ( or emulator after Monkey test)

adb –d reboot

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

How to make your device REMOTE (real device)?

* Set your Device and Laptop/PC to the SAME network
* Check on your Device : Settingsabout devicestatus ( write down )

This is a manual way to figure out what your IP is

But you may use adb commands to do the same.

adb -d shell netcfg

*( mine is 192.168.4.238 – please note your IP addr is different from mine, but I’m using my own numbers. Please adjust your command line to match your IP )*

or

adb -d shell ip addr

1. adb -d usb ( restarting in usb mode)

2. adb -d tcpip 5556 ( assigning port 5556 )

( checking if this port is available to use. Receive message "restarting in TCP mode port: 5556")

3. adb -d connect 192.168.4.243:5556  ( "marry” your IP address to the port 5556)

After receiving a message "connected to 192.168.4.243:5556"

Disconnect USB cable from computer and type:  adb devices

Your Device is now listed:

192.168.4.243:5556      device

Lets uninstall our APP from remote device ( you may use any APP )

adb -s 192.168.4.243:5556 uninstall “com.united.mobile.android”

and Install it back:

adb -s 192.168.4.243:5556 install “com.united.mobile.apk”

adb -s 192.168.4.243:5556 reboot

please note that you need to connect your usb cable back, because after reboot your device is NO LONGER remote

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

kill garbage from remote after you rebooted your device, connected usb cable, typed adb devices and see your device as real again.

your remote device might be listed as offline

adb -s yourip:5556 kill-server