**Session 5 :ANDROID STUDIO today :)**

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

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

**( if not go to SettingsAbout DeviceTap BUILD # 7 times**

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

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

**3. OPEN ANDROID STUDIO**

**4. check your devices if “TRUST THIS COMPUTER” notification appears -CONFIRM YES**

**5. Open Emulator**

**6. type : adb devices**

**7. if you have real device , then you will see both : real and emulator**

Lets install manually an app that is not in a Market Place (from APK folder on your Google Drive ( see instructions and task )

**PLEASE RENAME United App to com.united.mobile.apk**

Your file name will be com.united.mobile.apk ( I will be using this apk name as an example)

adb –d install com.united.mobile.apk

adb –d install < yourappnamegoeshere.apk>

 ( or to -e –emulator )this name is ONLY valid during an installation....Forget about this . *You can also rename it to anything like baby.apk - doesn’t matter. Because app information is stored in .apk and not in the name itself)*

Install APP if multiple devices attached: you may use -d or -e

In this document some of the devices are separated by serial number or -d, or -e.

adb –d install com.united.mobile.apk ( to a real device )

adb –e install com.united.mobile.apk ( to an emulator )

**OR**

adb -s emulator-5554 install com.united.mobile.apk ( or “com.united.mobile.apk” )

adb -s yourdevice#  install com.united.mobile.apk ( or “com.united.mobile.apk” )

*How to get package info:*

adb shell ( or adb –d or adb –e, or adb –s yourdevice# )

adb –d shell ( on my side, because I have two devices – real and emulator )

pm list packages  ( list all packages on your device)

pm list packages -f  ( will give you a full path where all the above packages being located)

pm list packages -f | grep united    ( you will get a package name for United APP )

pm list packages -f | grep lingocard    ( you will get a package name for United APP )

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

How to Collect LOGS from Real Device

6703cab3        device

 (this is my Real Device unique serial number. Your number is different)

emulator-5554   device

it’s recommended to clean up buffer before running each set of logs .

adb -d logcat -c

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

1. From the device itself:  ( to stop madness Ctrl+C)

adb -d logcat   ( it's going to run ALL logs possible )

adb –d logcat \*:E  ( for error logs only )

adb -d logcat \*:F  ( for fatal logs only )

adb -d logcat \*:I  ( for information logs only )

adb -d logcat \*:V  ( for verbose logs only )

adb -d logcat \*:D  ( for debugging logs only )

adb -d logcat \*:W  ( for warning logs only )

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

clean up buffer : adb –d logcat -c

To simultaneously collect logs from Device and place it into \*.txt

adb -d logcat \*:E >C:/a/online3.txt

clean up buffer : adb –d logcat -c

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Collecting Logs from APP

can use United or LingoCard in this exercise.

New command that is very presizely collecting logs. You need to consistenly working on your app

adb -d logcat \*:E >C:/a/online4.txt | findstr com.united.mobile.android

 ( OR com.lingocard.lingocard)

( sometime it's not working on Win7, but you can use the grep command below )

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

adb -d logcat \*:E >C:/a/online4.txt | adb -d shell grep com.united.mobile.android

 ( OR com.lingocard.lingocard)

type CTRL+C to stop collecting logs and go check \*.txt

You are welcome to give any name to .txt file

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx