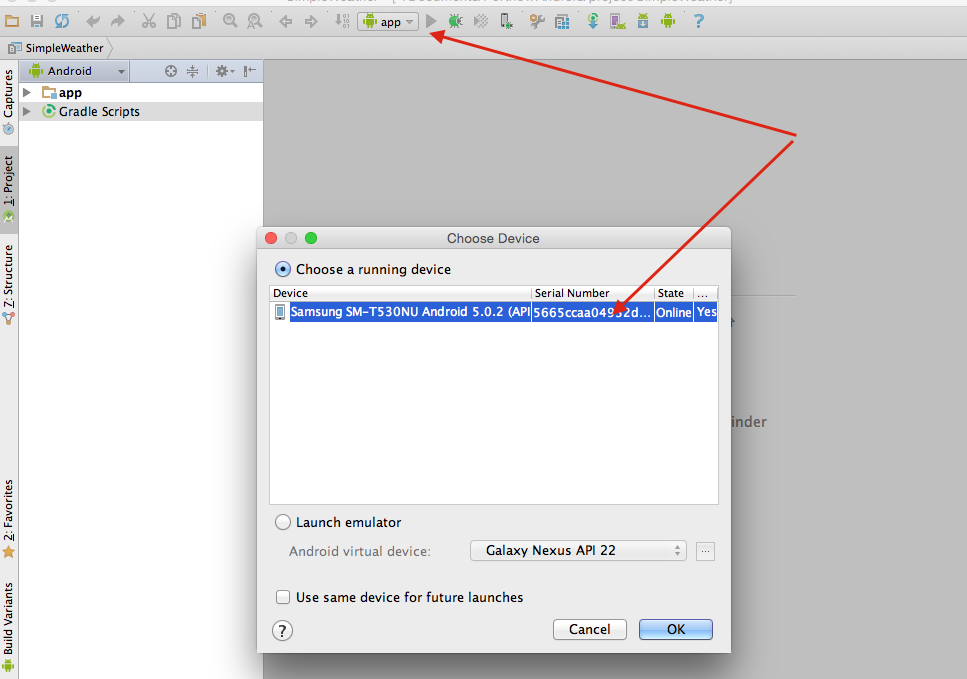
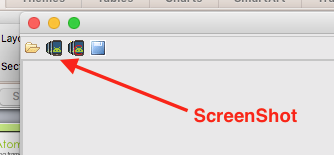
Exercises

1. Exercise - Enable USB debugging on the device
   1. Find out what Android version is running on your device
      1. Go to Settings>About phone/tablet
      2. Look under Android Version
   2. If Android less then 4.2
      1. Return to previous screen and go to Developer Options
      2. Enable USB debugging
   3. If Android more then 4.2
      1. Go to Settings>About phone
      2. Tap Build number seven times
      3. Return to previous screen and go to Developer Options
      4. Enable USB debugging
2. Exercise – Enable Unknown Sources Settings
   1. Settings>Security>Device Administrators
   2. Enable Unknown Sources
3. Exercise – check if your device is visible
   1. ./adb devices

NOTE: if you are on Windows machine and still don’t see your device, check if your device driver is installed

* go to ControlPanel>DeviceManager
* find the section for your Device driver
* check if it is installed or need update
* see more info on this matter at the end of this document

1. Exercise – build .apk with target ->your device
   1. Open Android Studio
   2. Attach your device
   3. Select Import Project
      1. If Android Studio is up and running, go File>Open
   4. Navigate to “android-UniversalMusicPlayer-master” project and open it
   5. In Android Studio, click on Green Arrow and specify your device as a target
   6. Verify the Simple Weather app is installed on your device
   7. 
2. Exercise - Install .apk to device
   1. Copy provided base.apk to ~/Library/Android/sdk/platform-tools
   2. Go to adb and run
      1. ./adb install base.apk
   3. Go to Device>Applications
   4. Find EventEzyGO
   5. Launch it and make sure it is working
3. Exercise - Get into your device
   * 1. Connect your device to computer
     2. Go to ~/Library/Android/sdk/platform-tools
     3. Run
        1. ./adb devices
     4. Find your device ID
     5. Run
        1. ./adb –s “device ID” shell
           1. (Cntr+D – to exit shell)
4. Exercise - Recording the device actions via adb
   1. Start Recording
      1. ./adb -s “device ID” shell screenrecord /sdcard/demo.mp4
   2. Make some actions on your device
   3. Stop Recording
      1. press Ctrl+C
   4. Retrieve the video
      1. ./adb pull sdcard/demo.mp4 ~/Desktop
   5. go to Desktop on your computer and verify you can play the video
5. Exercise – Deleting the file inside device
   1. Get inside your device
      1. adb shell
   2. navigate to sdcard
      1. cd /sdcard
   3. see the content of sdcard
      1. ls
   4. locate demo.mp4 and remove it
      1. rm demo.mp4
   5. verify demo.mp4 file is gone
      1. ls
6. Exercise – Recording the device action with debug output
   1. Launch test app (Universal Music Player)
   2. Get inside shell of your device
      1. ./adb shell
   3. Start recording using “verbose” option
      1. screenrecord --verbose /sdcard/demo1.mp4
   4. overview debug output in shell while you are exercising the app
7. Exercise – Recording the device action via Android Studio
   1. Build and start Simple Weather or install SimpleWeather apk to your device
   2. Connect your device via adb
   3. In Adroid Studio click on Android tab at the bottom of the window
   4. Find Green Arrow on the left side of the window
   5. Start using the SimpleWeather app on your device
   6. Create recording (by clicking on Green Arrow button) and save it to your Desktop
8. Exercise – find PID and Package Name
   1. Launch DDMS (Tools>Android>Android Device Monitor)
   2. Find PID and Package Name for Universal Music Player
9. Exercise – Uninstall .apk (application)
   1. First you need to find the package name for your app
      1. Open DDMS (Android Device Monitor)
      2. On the device start running the application (for example, Universal Music Player)
      3. Got to DDMS>Devices and find the package name for Universal Music Player
      4. go to adb directory and run
         1. ./adb uninstall <app package name>
10. Exercise – stop app via adb
    1. First re-install the app via adb
    2. stop app via PID
       1. ./adb shell kill <PID>
    3. re-install the app again
    4. stop app via Package Name
       1. ./adb shell am force-stop <PACKAGE>
11. Exercise - running Monkey
    1. Connect your device to computer
    2. Navigate to adb
    3. If you have several devices attached, find device ID via -> adb devices
    4. Find calculator package name
       1. ./adb shell pm list packages calculator
          1. the name might varies depending on the device, for example, com.android.calculator2 or com.sec.android.app.popupcalculator
          2. pm means ‘package manager’
    5. if you have several devices attached, run
       1. ./adb -s “device ID” shell monkey -p “calculator package name” –v 1000
    6. if you have only one device
       1. ./adb shell monkey –p “calculator package name” –v 1000
          1. Each **-v** on the command line will increment the verbosity level. Level 0 (the default) provides little information beyond startup notification, test completion, and final results. Level 1 provides more details about the test as it runs, such as individual events being sent to your activities. Level 2 provides more detailed setup information such as activities selected or not selected for testing.
          2. **–p** ->If you specify one or more packages this way, the Monkey will *only* allow the system to visit activities within those packages. If your application requires access to activities in other packages (e.g. to select a contact) you'll need to specify those packages as well. If you don't specify any packages, the Monkey will allow the system to launch activities in all packages. To specify multiple packages, use the -p option multiple times — one -p option per package.
          3. **-s** -> Seed value for pseudo-random number generator. If you re-run the Monkey with the same seed value, it will generate the same sequence of events:
          4. *./adb shell monkey –p “calculator package name” –s 777 –v 1000*
          5. you can redirect the output to txt file:
          6. *./adb shell monkey –p “calculator package name” –v 1000 > test.txt*
    7. observe the result on the device and in LogCat
12. Exercise – find crash with Monkey in Universal Music Player
    1. Launch Universal Music Player
    2. Start Monkey with 50 000 random events to be sent
    3. Observe result
    4. Collect log
13. Exercise – creating and retrieving Screenshot
    1. Make a Screenshot via your device
       1. Re-install Universal Music Player
       2. Start running it
       3. Press at the same time Power+VolumeDown buttons
       4. Go to adb and run
          1. ./adb pull /sdcard/Pictures/Screenshots ~/Desktop
       5. Go to Desktop on your computer and verify the screenshot is there
14. Exercise – creating Screenshot via DDMS
    1. Build and run Simple Weather app
    2. Connect your device via adb
    3. Open DDMS
    4. Click on Camera button and get the Screenshot
    5. Save the Screenshot
15. Exercise – creating Screenshot with UI Automator Viewer
    1. On your computer, on command line go to
       1. ~/Library/Android/sdk/tools
       2. run
          1. ./uiautomatorviewer
    2. Connect your device
    3. Start any app
    4. Go to your computer and click on “Device screenshot” button in the top left corner
    5. Observe Screenshot in the Viewer
    6. Click on Save button on ToolBar
    7. Navigate to the folder where you saved it and verify .png file is created
16. Exercise – Analyzing app’s UI components
    1. Make sure your device is connected
    2. Launch UI Automator Viewer
       1. Go to ….Android/sdk/tools
       2. From tools directory, run the following command:
          1. On Mac -> ./uiatomatorviewer
          2. On Win -> uiatomatorviewer
    3. Launch Universal Music Player on your device
    4. Inside UI Automator Viewer make a screen shot of your device screen
    5. 
    6. Find the ID Resource-ID for 5 UI elements

QA Test task for Tester candidates - My Blossom via Skype

if you already have Skype installed on your device, Uninstall it

connect your device to computer

cd adb directory

run ./adb uninstall com.skype.raider

1) Install Android SDK

2a) Connect an Android phone via USB to the computer

2b) If you do not have an android phone, you can use  the emulator that comes with the SDK

3) Verify that the device is connected and is recognized by the android sdk

4) Install your chatting app of preference (Skype, Qik, Snapchat...) using ADB commands in a CMD

to get Skype apk, go to <http://www.androidapksfree.com/apk/skype-apk-latest-version-download-free/>

scroll down to “Download Skype apk v5.4”

click on [Download APK from secure source](http://mars.androidapksfree.com/wp-content/uploads/2015/06/com.skype_.raider-v5.4.0.5871-84154095-Android-4.0.apk)

5) Start collecting logs, and save them to a file "YOURNAME\_test\_log.txt"

6) Sign into the app

7) Place a call/video to your preferred contact

8) Send a chat message

9) Provide a screen shot while the video call is ongoing (Hint DDMS)

10) Stop logging

11) So far I hope you have had enough time to find at least 1 bug in the app (Sad but true :( ) so create a bug report and add it to your notes.

12) Send me a list of test cases that you would create to cover the login/verification flow of the app

13) Choose one of them and write down the whole test case.

14) Send me via email all the files related to the test task.

\*When asked to use ADB, please write which is the command you used as you typed

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Usefull info:

Installing USB Driver

### Windows 8

<http://farmtek.net/win8_driver_fix.html>

### Windows 7

To install the Android USB driver on Windows 7 for the first time:

1. Connect your Android-powered device to your computer's USB port.
2. Right-click on Computer from your desktop or Windows Explorer, and select **Manage**.
3. Select **Devices** in the left pane.
4. Locate and expand Other device in the right pane.
5. Right-click the device name (such as Nexus S) and select **Update Driver Software**. This will launch the Hardware Update Wizard.
6. Select **Browse my computer for driver software** and click **Next**.
7. Click **Browse** and locate the USB driver folder. (The Google USB Driver is located in <sdk>\extras\google\usb\_driver\.)
8. Click **Next** to install the driver.

Or, to upgrade an existing Android USB driver on Windows 7 with the new driver:

1. Connect your Android-powered device to your computer's USB port.
2. Right-click on Computer from your desktop or Windows Explorer, and select **Manage**.
3. Select **Device Manager** in the left pane of the Computer Management window.
4. Locate and expand Android Phone in the right pane.
5. Right-click on Android Composite ADB Interface and select **Update Driver**. This will launch the Hardware Update Wizard.
6. Select **Install from a list or specific location** and click **Next**.
7. Select **Search for the best driver in these locations**; un-check **Search removable media**; and check **Include this location in the search**.
8. Click **Browse** and locate the USB driver folder. (The Google USB Driver is located in <sdk>\extras\google\usb\_driver\.)
9. Click **Next** to upgrade the driver.

Vendor ID is located in C:\Users\<username>\.android\adb\_usb.ini

Format of this file

1

Vendor ID

Recourse - <http://developer.android.com/tools/extras/oem-usb.html>

Useful links:

PCI Vendor List -

<http://www.pcidatabase.com/vendors.php?sort=id>

Motorola USB driver

<https://motorola-global-en-uk.custhelp.com/app/answers/prod_answer_detail/a_id/89881>

Toshiba USB driver

<http://support.toshiba.com/sscontent?docId=4001814>

Samsung USB driver

<http://www.samsung.com/us/support/downloads>

HTC USB driver

<http://www.htc.com/us/search/?submit=&q=usb+driver>

### USB Vendor IDs

|  |  |
| --- | --- |
| **Company** | **USB Vendor ID** |
| Acer | 0502 |
| ASUS | 0b05 |
| Dell | 413c |
| Foxconn | 0489 |
| Fujitsu | 04c5 |
| Fujitsu Toshiba | 04c5 |
| Garmin-Asus | 091e |
| Google | 18d1 |
| Haier | 201E |
| Hisense | 109b |
| HTC | 0bb4 |
| Huawei | 12d1 |
| Intel | 8087 |
| K-Touch | 24e3 |
| KT Tech | 2116 |
| Kyocera | 0482 |
| Lenovo | 17ef |
| LG | 1004 |
| Motorola | 22b8 |
| MTK | 0e8d |
| NEC | 0409 |
| Nook | 2080 |
| Nvidia | 0955 |
| OTGV | 2257 |
| Pantech | 10a9 |
| Pegatron | 1d4d |
| Philips | 0471 |
| PMC-Sierra | 04da |
| Qualcomm | 05c6 |
| SK Telesys | 1f53 |
| Samsung | 04e8 |
| Sharp | 04dd |
| Sony | 054c |
| Sony Ericsson | 0fce |
| Sony Mobile Communications | 0fce |
| Teleepoch | 2340 |
| Toshiba | 0930 |
| ZTE | 19d2 |