Installation DroydRunner

Sous linux (Débian, Rasbian, Ubuntu)

Installation

* Mise à jour
* Ajout des outils adb d’android
* Installation de Git
* Installation des lib Python

sudo apt-get update

sudo apt-get install android-tools-adb androidsdk-uiautomatorviewer

sudo apt-get install git

sudo apt-get install python-pip

Activation mode développeur + Débogage USB sur le mobile,

* Cliquez sur « Paramètres »,
* Ouvrir « À propos de l’appareil ».
* Tapotez par sept fois le « Numéro de Build » (éventuellement dans « Infos logiciel »)
* Pour débloquer les Options pour les développeurs. Revenez dans « Paramètres »,
* « Options de développement » est disponible.
* Cliquez sur « Options de développement »,
* Activez « Débogage USB ».

Pour vérifier utiliser la commande suivante pour voir les appareils branchés en USB :

adb devices

Intallation de Droydrunner (voir fichier ZIP joint)

Pour un mobile local : (droydserver.robot\_plugin.Pilot)

cd /tmp

unzip Droydrunner\_vX.X.X.zip -d DroydSetup

cd DroydSetup

sudo pip2.7 install -r requirements\_server.txt

sudo python2.7 setup\_server.py install

Pour un mobile distant : (droydclient.robot\_plugin.Pilot)

cd /tmp

unzip Droydrunner\_vX.X.X.zip -d DroydSetup

cd DroydSetup

sudo pip2.7 install -r requirements\_client.txt

sudo python2.7 setup\_client.py install

Lancer une application depuis le pc linux : (ici la calculatrice)

adb shell am start -n "com.sec.android.app.popupcalculator/.Calculator"

pour chercher la ligne correspondant à une application :

adb shell dumpsys activity | grep Intent

# les lignes retournées

Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10200000 cmp=com.sec.android.app.popupcalculator/.Calculator bnds=[371,1184][719,1616] (has extras) }

Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10200000 cmp=com.sec.android.app.popupcalculator/.Calculator bnds=[371,1184][719,1616] (has extras) }

Pour étudier l’interface graphique d’une application on utilise l’utilitaire graphique :

uiautomatorviewer &

On peut interagir à partir des « resources-id » pour les events comme Clicks

Voici un exemple de script DroydRunner sur la calculatrice

\*\*\* Settings \*\*\*

Library droydserver.robot\_plugin.Pilot

Library Collections

Library String

\*\*\* Variables \*\*\*

${id\_mobile} XXXXXXXXXXX

# lancement : export RF\_HUB\_URL=http://X.X.X.X:5000

# : pybot simplecall.robot

\*\*\* Test Cases \*\*\*

Test

Builtin.Log To Console 'Test'

Open Session ${id\_mobile}

Execute Adb Shell Command ${id\_mobile} cmd=am start -n "com.sec.android.app.popupcalculator/.Calculator"

Builtin.Sleep 1

Click ${id\_mobile}

resourceId=com.sec.android.app.popupcalculator:id/bt\_08

Click ${id\_mobile}

resourceId=com.sec.android.app.popupcalculator:id/bt\_add

Click ${id\_mobile}

resourceId=com.sec.android.app.popupcalculator:id/bt\_03

Click ${id\_mobile}

resourceId=com.sec.android.app.popupcalculator:id/bt\_equal

Builtin.Sleep 1

${objMsg} Get Object ${id\_mobile} resourceId=com.sec.android.app.popupcalculator:id/txtCalc

${contentMsg} Get Info Of Object ${id\_mobile}

obj=${objMsg}

${textMsg} evaluate ${contentMsg}.get("text")

Builtin.Log To Console ${textMsg}

Close session