Introduction to Scripting Apps on Android

Scott A. Thisse
April 2, 2014
Introduction to Scripting Apps on Android

- The most common way of doing scripting on Android is via a project called:

  Scripting Layer for Android (SL4A)
Scripting Layer for Android (SL4A)

• Acts as a host app that allows scripting

• Interpreters for several languages:
  
  BeanShell (Java), JRuby (working?), Lua, PHP, Perl, Python, Rhino (JavaScript) Android Shell, Tcl (?)
Scripting Layer for Android (SL4A)

• Built as a client-server solution
• Server exposes Android API methods in abstractions called *facades*
• Client communicates with server via RPC/JSON
• This approach provides for better security
Scripting Layer for Android (SL4A)

- Allows on-device editing and execution
- Allows remote execution via PC (using Eclipse/PyDev, others)
- Allows for packaging into APKs
Scripting Layer for Android (SL4A)

- Alternative scripting solutions:
  - JRuby and Ruboto
  - Embedded interpreter (CPython)
  - QPython (based on Qt)
Installing SL4A

- Download from *android-scripting* project on Google Code Projects
- Latest version is *SL4A Release 6*
- Install the APK and run the app
Installing SL4A

• Only an Android shell is provided by default

• Interpreters can be installed via Menu > View > Interpreters > Menu > Add

• But we'll do our installs manually
Installing SL4A

• Install Python 2.6.2 and 3.2.2 from the Python-for-Android project

• Latest stable Python 2 is Py4A Release 5

• Latest Python 3 is Python3 APK R6
Installing SL4A

- Download, install, and run both APKs
- *Browse Modules* to see other modules to install (such as Twisted)
- *Import Modules* to install downloaded modules
Using SL4A

• Run SL4A app

• Clicking on a script displays:
  • Terminal – runs script in a terminal window
  • Cog Wheel – runs script in the background
  • Pencil – opens script in an editor
  • Diskette – presents rename dialog
  • Right-arrow sign – opens docs window
Using SL4A

- In the editor, menu has several buttons

- *API Browser* is interesting:
  - Press to get info
  - Long-press to get insert template, run a wizard, or get help
Using SL4A

- Android API exposed via calls through object named `android.Android()`

- Results always returned in a dictionary:
  - `id`: unique, increasing number
  - `result`: return value (or null)
  - `error`: error description (or null)
Using SL4A

Here are some of the facades:

ActivityResultFacade, AndroidFacade, ApplicationManagerFacade, BatteryManagerFacade, BluetoothFacade, Bluetooth, CameraFacade, CommonIntentsFacade, ContactsFacade, EventFacade, EyesFreeFacade, LocationFacade, MediaPlayerFacade, MediaRecorderFacade, PhoneFacade, PreferencesFacade, SensorManagerFacade, SettingsFacade, SignalStrengthFacade, SmsFacade, SpeechRecognitionFacade, TextToSpeechFacade, ToneGeneratorFacade, UiFacade, User, WakeLockFacade, WebCamFacade, WifiFacade
Using SL4A

Let's look at some apps:

- Barcode scanning
- Starting activities
- Battery stats
- Displaying dialogs

- Sending email
- Display Web Views
- Listing installed apps
- Taking Pictures
Other Topics

Some other things to discover:

- Remote execution/debugging
- Packaging into APKs
- QPython (built on top of SL4A, it looks like it also does PyGame, etc.)
Q & A

Questions?
References

