Android Application Development – A Kickstart

Syed Basit Ali Jafri
Nasir Ibrahim
Agenda

- Introduction
- Why Android
- Android Development – How to
Phones and Smartphones
Operating System?

- Windows
- Linux
- Mac
- Symbian
- Windows-Mobile
- BlackBerry OS

Android!
Why Android?

Mobile OS

Google
Android Facts!

- Linux based open source OS
- Supported by Google and Open Handset Alliance
- Java programming for developers
- Already 15 phones available
Local Android Development Efforts
Who we are?

Mobile apps @ Pepper.pk
Android Development
Android Architecture

Applications
Built-in (phone, contacts, browser), Third-party/Custom

Application frameworks
Telephone Mgr, Location Mgr, Notification Mgr, Content providers, Windowing, Resource Mgr, etc.

Libraries
Graphics, media, database, WebKit, etc.

Android runtime
Dalvik Virtual Machine

Linus Kernel
Power, File system, drivers, process, management, etc.
Application Framework

- Developers
  - can build extremely rich and innovative applications
  - have full access to framework APIs
Underlying all applications is a set of services and systems, including:
- Views
- Content Providers
- Resource Manager
- Notification Manager
- Activity Manager
Application Components

- no main() function
- There are four types of components:
  - Activities
  - Services.
  - Broadcast Receivers
  - Content providers
Step 1 – Downloads:

- Eclipse
  http://www.eclipse.org/downloads/

- Andriod SDK
  http://developer.android.com/sdk/
Development Kickstart

- **Step 2 – Configure Eclipse and SDK**

- This includes
  - Configure Eclipse Plugin for Android
  - Configure SDK with Eclipse
Configure Eclipse Plugin – Step 1
Configure Eclipse Plugin Step 2
Configure Eclipse Plugin – Step 3

Add Site

Location: https://dl-ssl.google.com/android/eclipse/
Configure Eclipse Plugin – Step 4

Open the 'Automatic Updates' preference page to set up an automatic update schedule.
Configure Android SDK

![Android SDK Configuration](image-url)
Hello world example
The **View** class serves as the base for subclasses called "widgets".

The **ViewGroup** class serves as the base for subclasses called "layouts".
# Common Layout Objects

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrameLayout</td>
<td>Layout that acts as a view frame to display a single object.</td>
</tr>
<tr>
<td>Galllery</td>
<td>A horizontal scrolling display of images, from a bound list.</td>
</tr>
<tr>
<td>GridView</td>
<td>Displays a scrolling grid of m columns and n rows.</td>
</tr>
<tr>
<td>LinearLayout</td>
<td>A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.</td>
</tr>
<tr>
<td>ListView</td>
<td>Displays a scrolling single column list.</td>
</tr>
<tr>
<td>RelativeLayout</td>
<td>Enables you to specify the location of child objects relative to each other</td>
</tr>
<tr>
<td>TableLayout</td>
<td>A tabular layout with an arbitrary number of rows and columns, each cell holding the widget of your choice.</td>
</tr>
</tbody>
</table>
Declaring a Layout

- `<xml version="1.0" encoding="utf-8"?>
  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView android:id="@+id/text"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello, I am a Button" />
  </LinearLayout>`
Menus

- Options Menu
  - Icon Menu
  - Expanded Menu
- Context Menu
- Submenu
Dialogs

- A small window that appears in front of the current Activity
  - Alert Dialog
  - Progress Dialog
  - DatePicker Dialog

- Custom dialog
  - create your own layout for the dialog window with layout and widget elements.

- Toast Notification is a message that pops up on the surface of the window
Security and Permissions

- Android is a multi-process system
- Most security is enforced at the process level
- All Android applications (.apk files) must be signed
- By default, no permissions are associated with it.
- AndroidManifest.xml file can be used to define permissions.
Location services supported by the device can be gained through the classes in the `android.location` package.

**LocationManager** is the central component of the location framework

`getSystemService(Context.LOCATION_SERVICE)`
Publish Applications

- All applications *must* be signed.
- Use self-signed certificates to sign applications.
- Standard tools — Keytool and Jarsigner
basit@pepper.pk
nasir@pepper.pk