Mobile – Android Intro

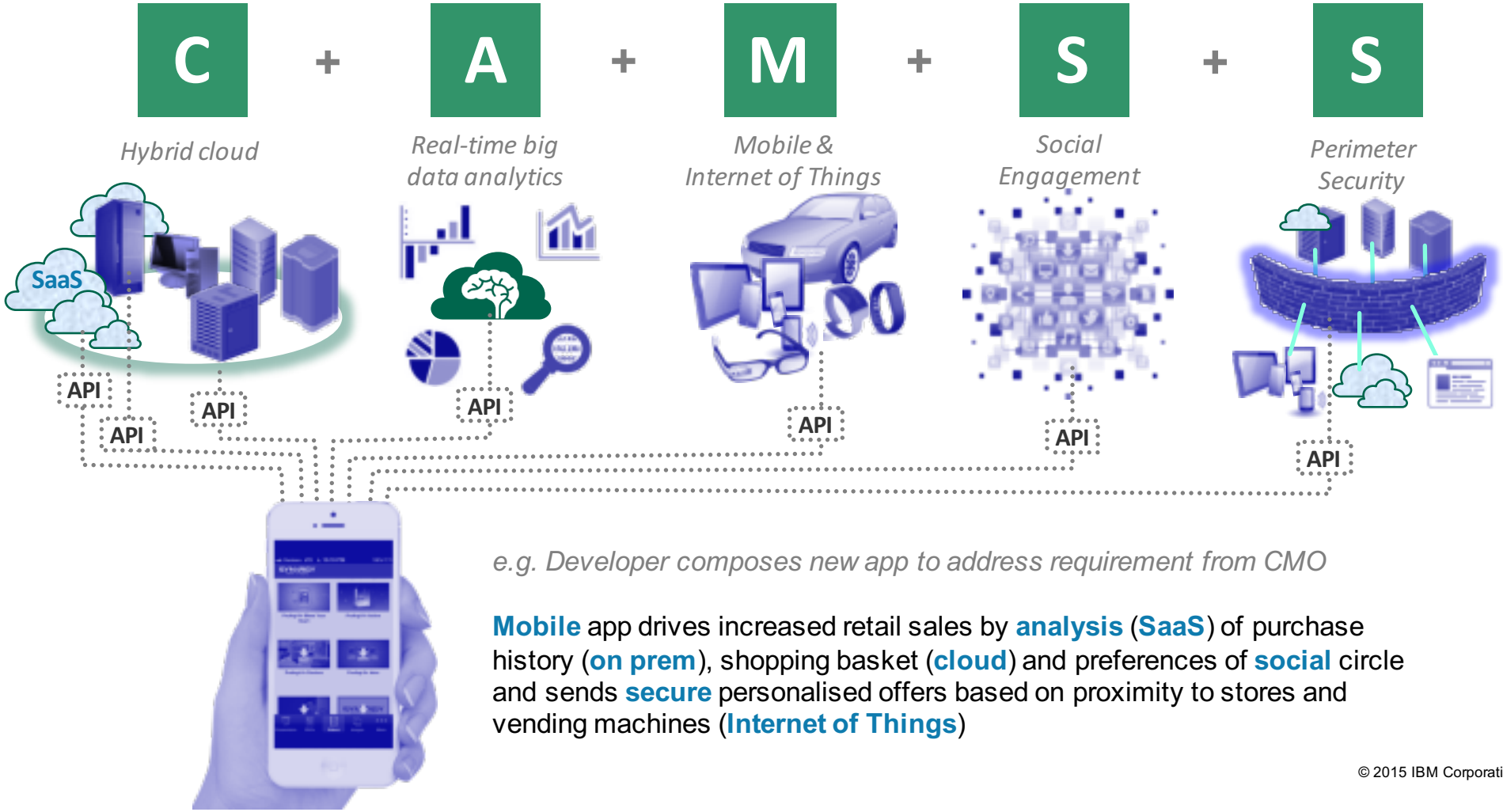
Presented by:

IBM EcoD

IBM **Cloud**

Mobile Cloud = Bringing the Real Power of CAMSS

- Customer's energy around CAMSS is not driven by 5 **distinct** parallel agendas
- Combining CAMSS in tandem **delivers agility** to create new business opportunities
- API-centric approach enable CAMSS to be **rapidly composed** as building blocks



e.g. Developer composes new app to address requirement from CMO

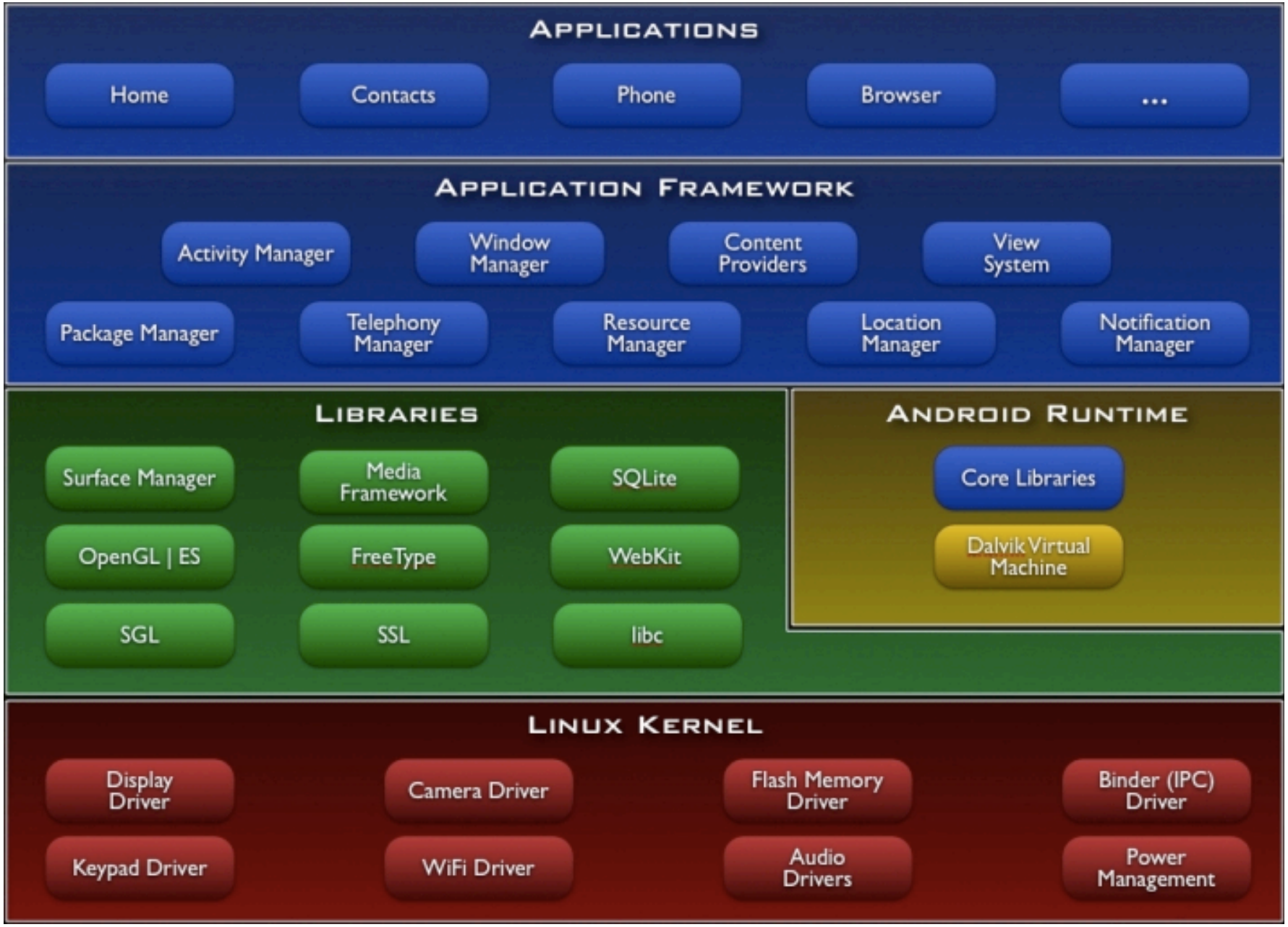
Mobile app drives increased retail sales by **analysis** (**SaaS**) of purchase history (**on prem**), shopping basket (**cloud**) and preferences of **social** circle and sends **secure** personalised offers based on proximity to stores and vending machines (**Internet of Things**)

© 2015 IBM Corporation

What is Android?

- A software stack for mobile devices that includes
  - An operating system
  - Middleware
  - Key Applications
- Uses Linux to provide core system services
  - Security
  - Memory management
  - Process management
  - Power management
  - Hardware drivers

© 2015 IBM Corporation



http://developer.android.com/guide/basics/what-is-android.html

Android Features

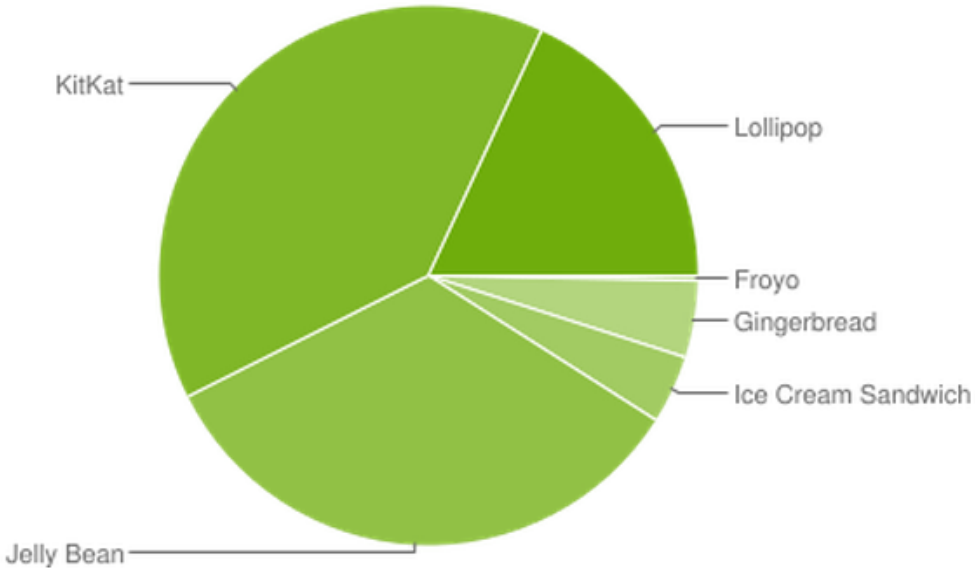
- **Application framework** enabling reuse and replacement of components
- **Dalvik virtual machine** optimized for mobile devices
- **Integrated browser** based on the open source WebKit engine
- **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- **GSM Telephony** (hardware dependent)
- **Bluetooth, EDGE, 3G, and WiFi** (hardware dependent)
- **Camera, GPS, compass, and accelerometer** (hardware dependent)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

http://developer.android.com/guide/basics/what-is-android.html

© 2015 IBM Corporation

Android Distribution Aug 2015

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	4.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	4.1%
4.1.x	Jelly Bean	16	13.0%
4.2.x		17	15.9%
4.3		18	4.7%
4.4	KitKat	19	39.3%
5.0	Lollipop	21	15.5%
5.1		22	2.6%



https://developer.android.com/about/dashboards/index.html

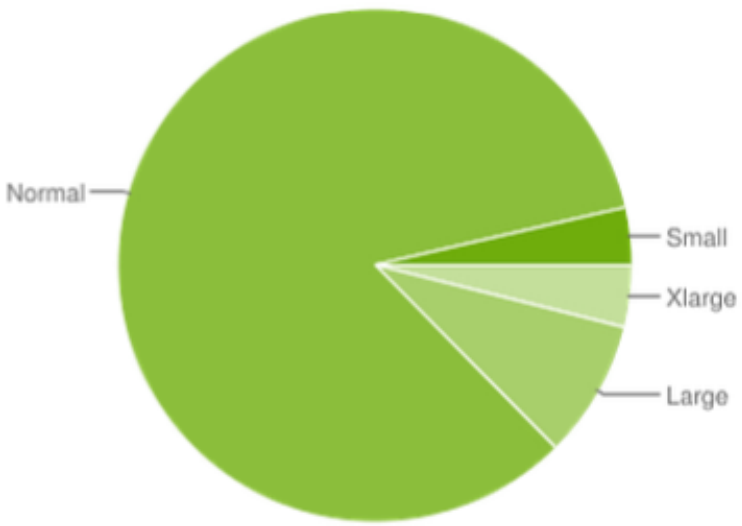
Data collected during a 7-day period ending on August 3, 2015.

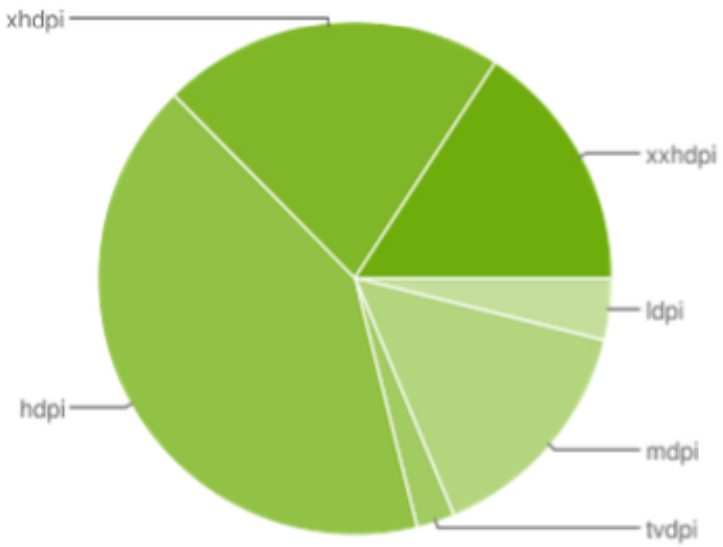
Any versions with less than 0.1% distribution are not shown.

© 2015 IBM Corporation

Screen Densities as of Aug 2015

	ldpi (~120dpi)	mdpi (~160dpi)	tvdpi (~240dpi)	hdpi (~320dpi)	xhdpi (~480dpi)	xxhdpi (~640dpi)	Total
Small	3.6%						3.6%
Normal		6.9%	0.1%	40.6%	20.4%	15.8%	83.8%
Large	0.3%	4.9%	2.3%	0.6%	0.6%		8.7%
Xlarge		3.0%		0.3%	0.6%		3.9%
Total	3.9%	14.8%	2.4%	41.5%	21.6%	15.8%	





Data collected during a 7-day period ending on August 3, 2015.

Any screen configurations with less than 0.1% distribution are not shown.

Lets talk about Android Runtime!!!

Android Runtime – Dalvik VM

- Subset of Java developed by Google
- Optimized for mobile devices (better memory management, battery utilization, etc.)
- Dalvik runs .dex files that are compiled from .class files
- Introduces new libraries
- Does not support some Java libraries like AWT, Swing

http://developer.android.com/reference/packages.html

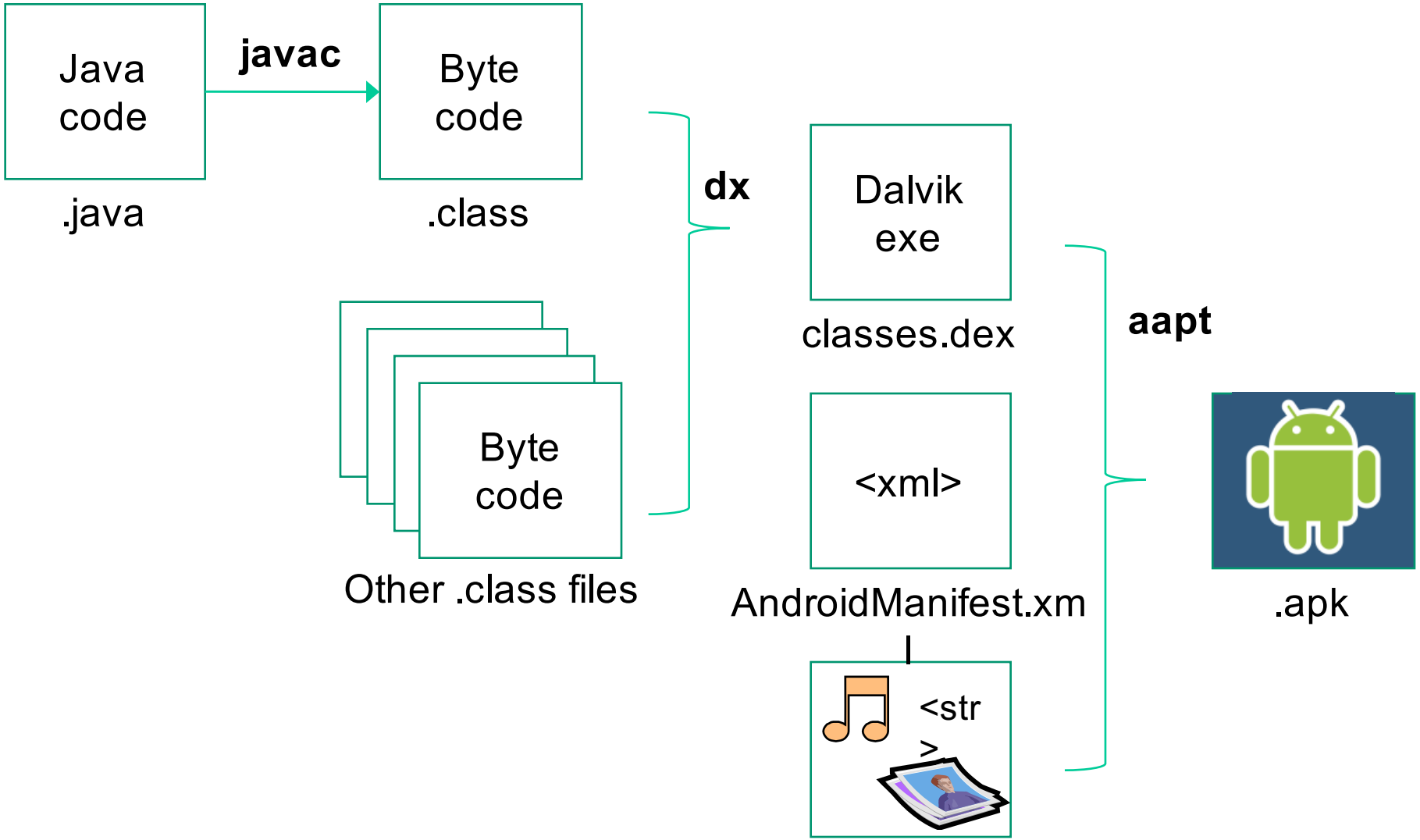
© 2015 IBM Corporation

Boxed Applications

- By default, each app is run in its own Linux process
  - Process started when app's code needs to be executed
  - Threads can be started to handle time-consuming operations
- Each process has its own Dalvik VM
- By default, each app is assigned unique Linux ID
  - Permissions are set so app's files are only visible to that app

© 2015 IBM Corporation

How to Produce a Android App?



© 2015 IBM Corporation

Emulator

© 2015 IBM Corporation

Emulator Basics

- Host computer's keyboard works
- Host's mouse acts as finger
- Uses host's Internet connection
- Other buttons work: Home, Menu, Back, Search, volume up and down, etc.
- Ctrl-F11 toggle landscape → portrait
- Alt-Enter toggle full-screen mode
- More info at http://developer.android.com/guide/developing/devices/emulator.html

© 2015 IBM Corporation

Emulator Limitations

- No support for placing or receiving actual phone calls
  - Simulate phone calls (placed and received) through the emulator console
- No support for USB connections
- No support for camera/video capture (input)
- No support for device-attached headphones
- No support for determining connected state
- No support for determining battery charge level and AC charging state
- No support for determining SD card insert/eject
- No support for Bluetooth
- No support for simulating the accelerometer
  - Use OpenIntents's Sensor Simulator

© 2015 IBM Corporation

Android Emulator (Android Virtual Device – AVD)

- Emulator is essential to testing app but is not a substitute for a real device
- Emulators are **Android Virtual Devices** (AVDs)
- Android SDK and AVD Manager allows you to create AVDs that target any Android API level
- AVD have configurable resolutions, RAM, SD cards, skins, and other hardware

© 2015 IBM Corporation

Labs

© 2015 IBM Corporation

Pre-Requisites

- Install and Setup Android Studio following steps in the document attached

Microsoft Word Document

- OR Follow the instructions → https://github.com/ibmecod/mobile-androidintro/blob/master/PreReqs.pdf

© 2015 IBM Corporation

Agenda – What are you going to learn?

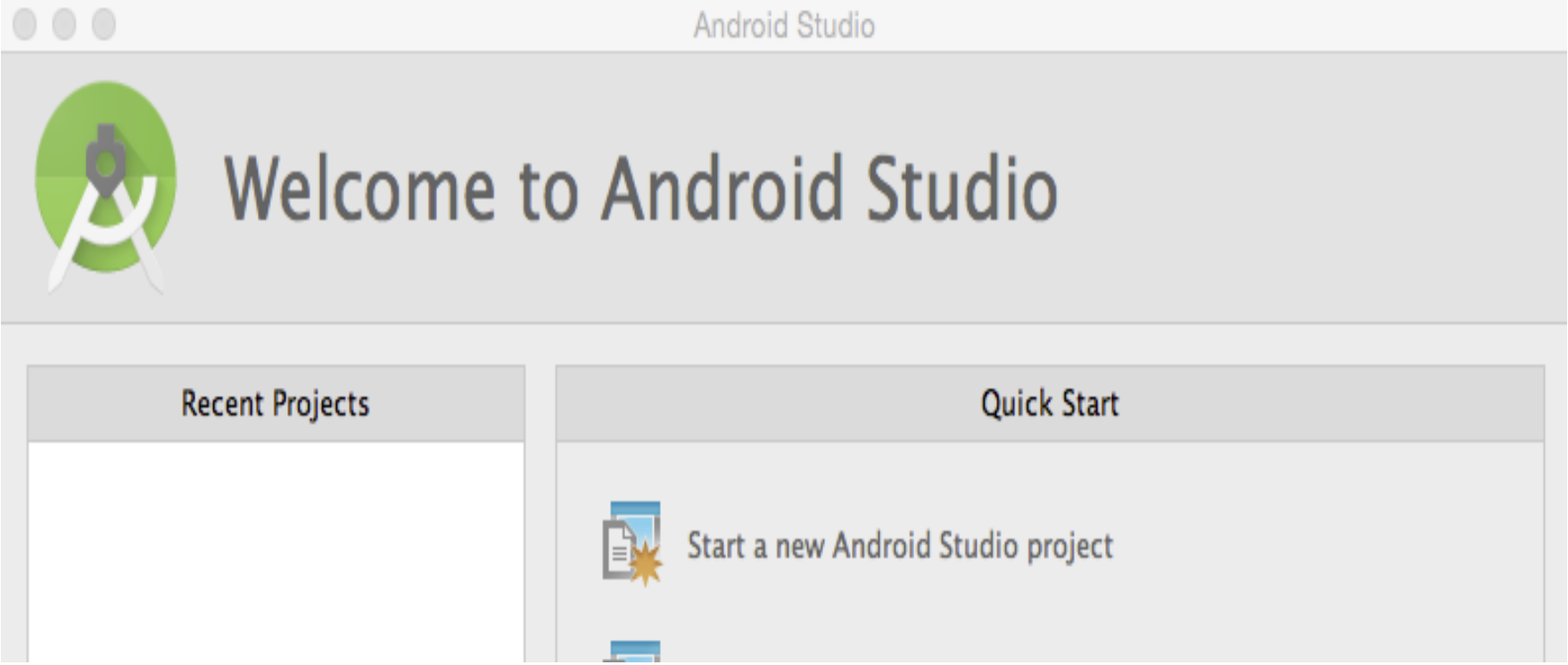
1. Create an Activity
2. Demonstrate resources created
3. Show the Activity lifecycle within the Android OS
4. Show the various debugging tools available
5. Show how to start one Activity from another

© 2015 IBM Corporation

Lab1 – Create your first Android Hello World Application

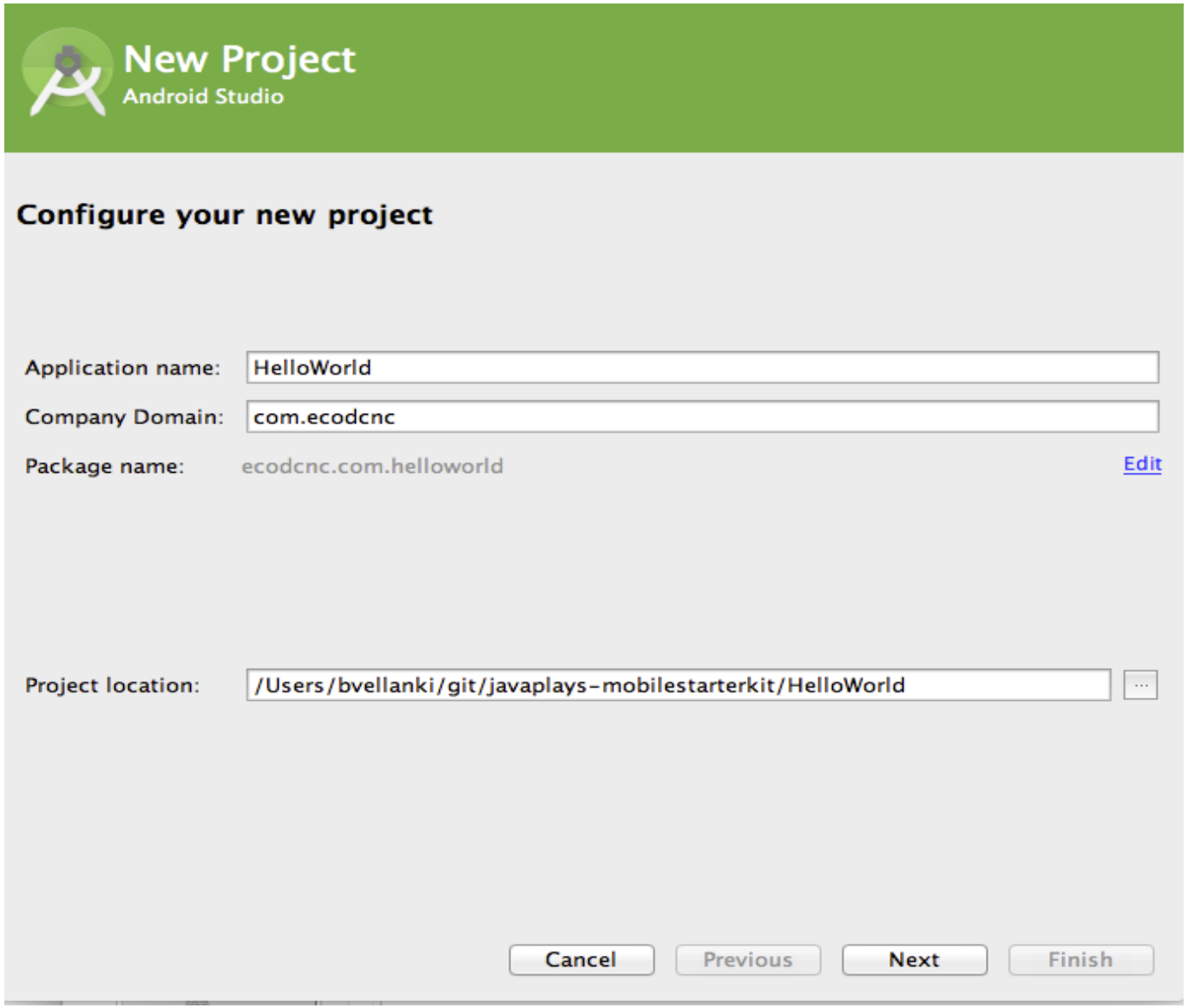
© 2015 IBM Corporation

Open Android Studio → Start New Project



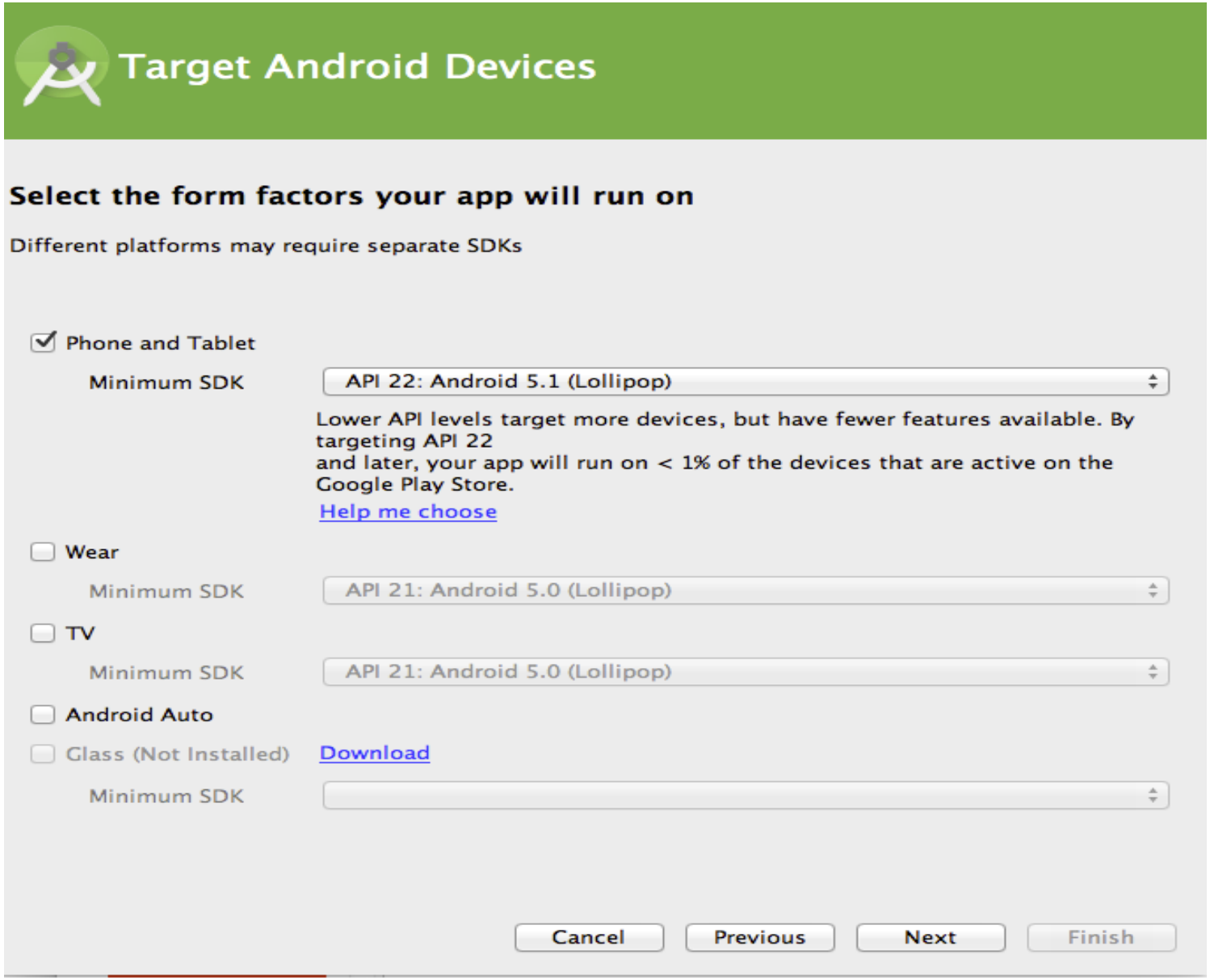
© 2015 IBM Corporation

Configure Android Project → Name your App



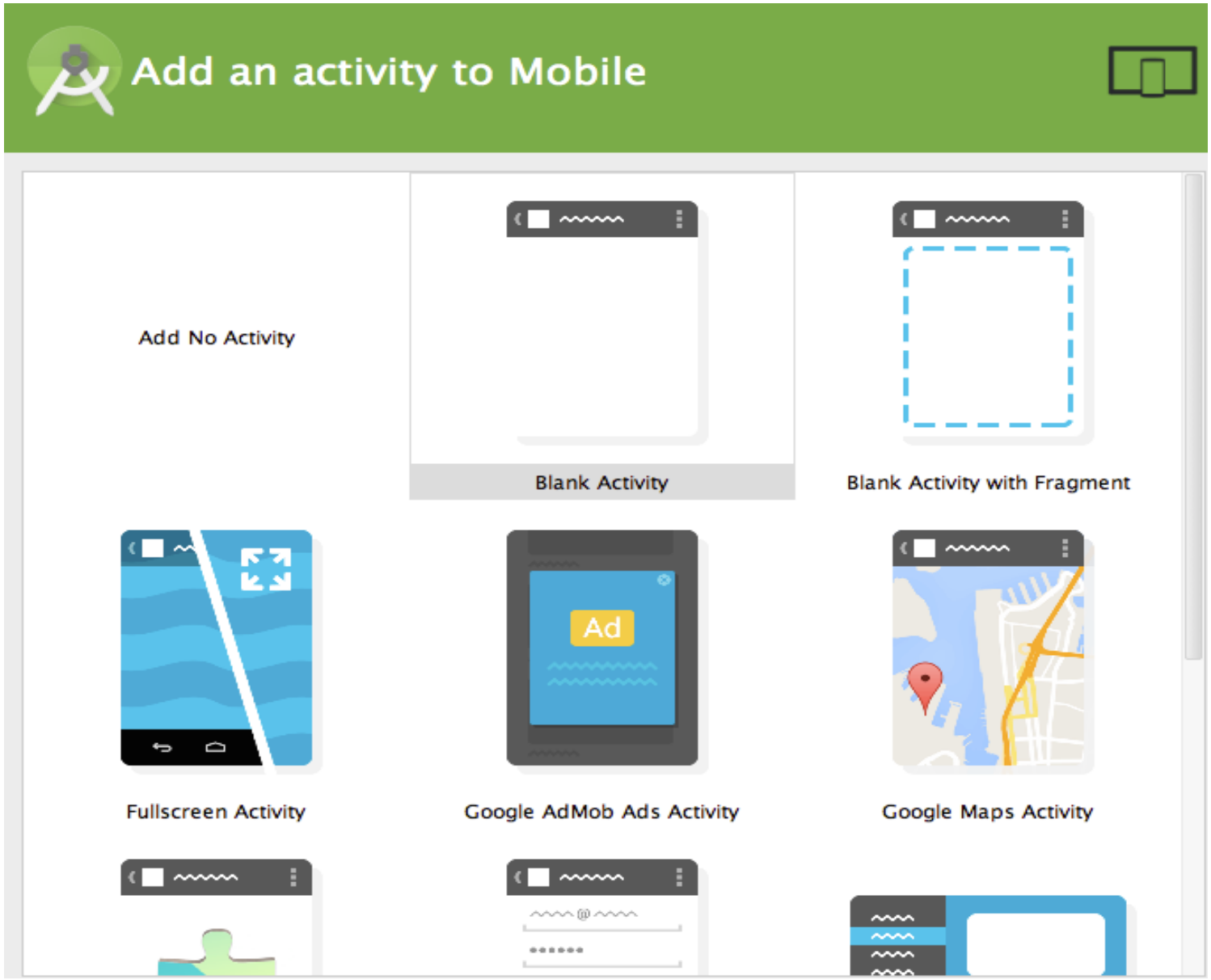
© 2015 IBM Corporation

Select an Android SDK for your app to use



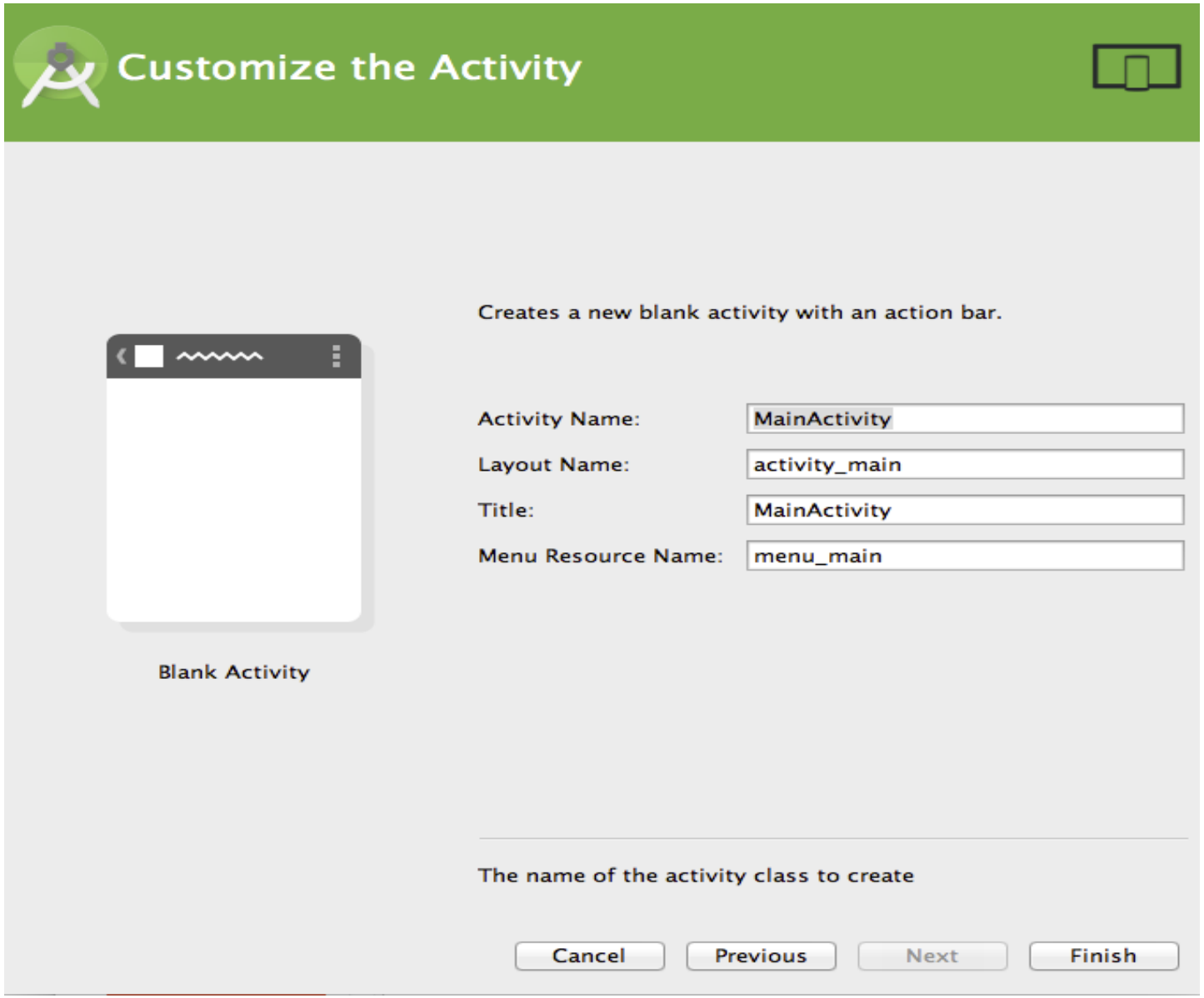
© 2015 IBM Corporation

Add an Activity to your App



© 2015 IBM Corporation

Create Main Activity – Name the Activity



© 2015 IBM Corporation

Launch the App

app
HelloWorld > app > src > main > res > layout > activity_main.xml

Android
app
manifests
java
res
drawable
layout
activity_main.xml
menu
mipmap
values
Gradle Scripts

MainActivity.java | activity_main.xml

Palette
Layouts
FrameLayout
LinearLayout (Horizonta
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
ProgressBar (Horizonta
SeekBar
RatingBar
Spinner
WebView
Text Fields
Plain Text
Person Name
Password
Password (Numeric)
E-mail
Phone
Postal Address
Multiline Text
Time
Date

Nexus 4 | AppTheme | MainActivity | 22

5:10
HelloWorld
Hello world!

Component Tree
Device Screen
RelativeLayout
TextView – @string/hello_world

Properties

layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveRegion	
accessibilityTraversalA	
accessibilityTraversalB	
alpha	
background	
backgroundTint	
backgroundTintMode	
clickable	☐
contentDescription	
elevation	
focusable	☐
focusableInTouchMode	☐
gravity	[]
id	
ignoreGravity	
importantForAccessibil	

Design | Text

Terminal | 6: Android | 0: Messages | TODO | 1 Event Log | Gradle Console

Success!!! Yahoo!!

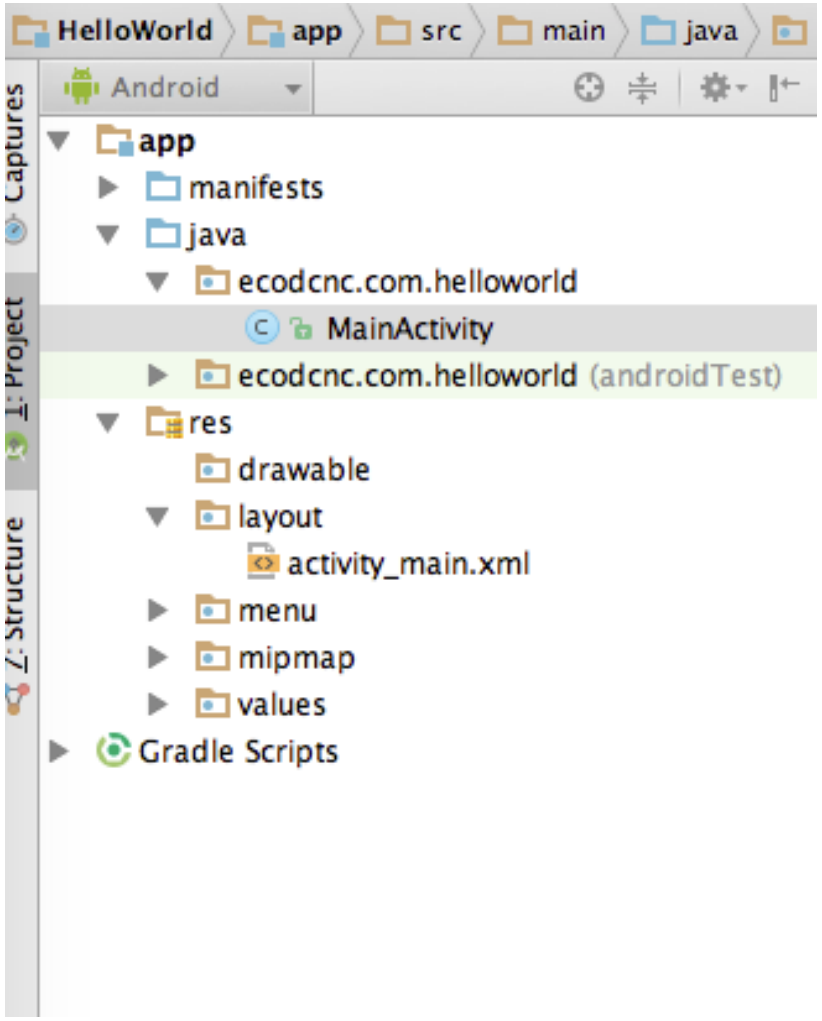
app
HelloWorld > app > src > main > res > layout > activity_main.xml
Android
app
manifests
java
res
drawable
layout
activity_main.xml
menu
mipmap
values
Gradle Scripts
MainActivity.java
activity_main.xml
Palette
Layouts
FrameLayout
LinearLayout (Horizonta
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
ProgressBar (Horizonta
SeekBar
RatingBar
Spinner
WebView
Text Fields
Plain Text
Person Name
Password
Password (Numeric)
E-mail
Phone
Postal Address
Multiline Text
Time
Date
Nexus 4
AppTheme
MainActivity
22
5:10
HelloWorld
Hello world!
Component Tree
Device Screen
RelativeLayout
TextView – @string/hello_world
Properties
layout:width match_parent
layout:height match_parent
style
accessibilityLiveRegion
accessibilityTraversalA
accessibilityTraversalB
alpha
background
backgroundTint
backgroundTintMode
clickable
contentDescription
elevation
focusable
focusableInTouchMode
gravity []
id
ignoreGravity
importantForAccessibil
Design
Text
Terminal
6: Android
0: Messages
TODO
1 Event Log
Gradle Console
Gradle build finished in 1s 706ms (a minute ago)
n/a
n/a
Context: <no context>

© 2015 IBM Corporation

Lab2 – Add Code to Your App

© 2015 IBM Corporation

Open HelloWorld Project → Java → MainActivity.java



© 2015 IBM Corporation

MainActivity.java

MainActivity.java × activity_main.xml ×

```java
package example.com.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView; // LAB 2 NEW

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.activity_main);  // LAB 2 COMMENTED OUT

        // LAB 2 THREE NEW LINES ADDED
        TextView textView = new TextView(this);
        textView.setText("Hello New World!");
        setContentView(textView);

    }
```

© 2015 IBM Corporation

Update MainActivity.java

- Add
  import android.widget.TextView;

- In the OnCreate Method
  1. Comment out THIS LINE
     // setContentView(R.layout.activity_main);

  2. Add the lines
     TextView textView = new TextView(this);
     textView.setText("Hello New World!");
     setContentView(textView);

© 2015 IBM Corporation

Launch Lab 2 – Hurray!

HelloWorld - [C:\Users\IBM_ADMIN\Desktop\rtpjavabootcamp\studioworkspace\Hello

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

app

HelloWorld > app > src > main > java > example > com > helloworld > MainActivity

Android

- app
  - manifests
    - AndroidManifest.xml
  - java
    - example.com.helloworld
      - MainActivity
    - example.com.helloworld (androidTest)
  - res
    - drawable
    - layout
      - activity_main.xml
    - menu
      - menu_main.xml

MainActivity.java × | activity_main.xml ×

```
package example.com.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView; // LAB 2 NEW

public class MainActivity extends AppCompatActivi

    @Override
    protected void onCreate(Bundle savedInstanceS
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.activity_main);

        // LAB 2 THREE NEW LINES ADDED
        TextView textView = new TextView(this);
        textView.setText("Hello New World!");
        setContentView(textView);
```

5554:Nexus_5_API_23

1:33

HelloWorld

Hello New World!

© 2015 IBM Corporation

Lab3 – Log and View

© 2015 IBM Corporation

Updated MainActivity.java – Screen shot

```java
package ecodcnc.com.helloworld;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.util.Log;


import android.widget.TextView;


public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.activity_main);

        Log.i(this.getClass().getName(), "Hello Main Activity Logging Started")


        TextView textView = new TextView(this);
        textView.setText("Hello New World!");
        setContentView(textView);

        Log.i(this.getClass().getName(), "Hello Main Activity Logging Ended");

    }
```

© 2015 IBM Corporation

Update MainActivity.java

- Add to Import

```
import android.util.Log;
import android.widget.TextView;
```

- ADD THESE LINES highlighted in green to OnCreate Method

```
Log.i(this.getClass().getName(), “TEXT1");

    TextView textView = new TextView(this);  // 3 lines
      already exists from lab2
    textView.setText("Hello New World!");
    setContentView(textView);

Log.i(this.getClass().getName(), “TEXT2");
```

© 2015 IBM Corporation

Run and Check Log Filter – Verify the Messages

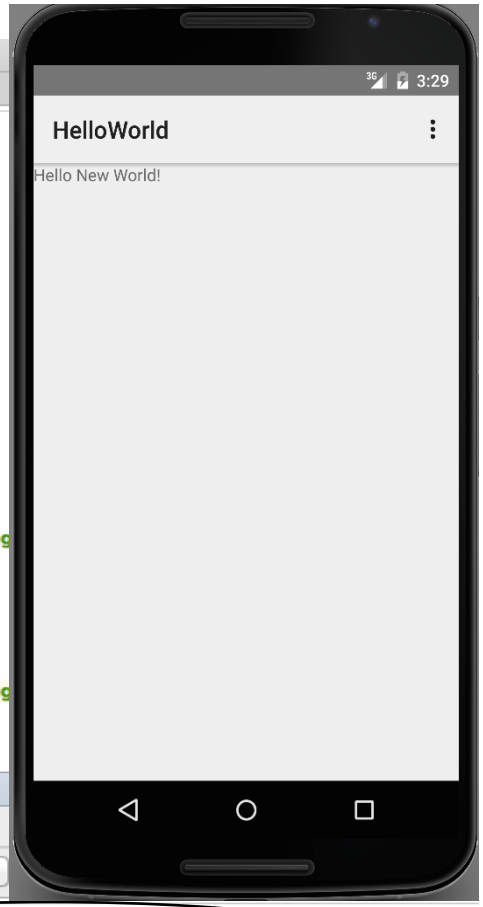
HelloWorld > app > src > main > java > ecodcnc > com > helloworld > MainActivity

Android

- app
  - manifests
  - java
    - ecodcnc.com.helloworld
      - MainActivity
    - ecodcnc.com.helloworld (androidTest)
  - res
    - drawable
    - layout
      - activity_main.xml
    - menu
    - mipmap
    - values
- Gradle Scripts

MainActivity.java × | activity_main.xml ×

```java
package ecodcnc.com.helloworld;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.util.Log;


import android.widget.TextView;


public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.activity_main);

        Log.i(this.getClass().getName(), "Hello Main Activity Logging


        TextView textView = new TextView(this);
        textView.setText("Hello New World!");
        setContentView(textView);

        Log.i(this.getClass().getName(), "Hello Main Activity Logging

    }
```



Android

Emulator Nexus_6_API_22 Android 5.1 (API 22) | ecodcnc.com.helloworld (1768)

logcat | ADB logs | Memory | CPU | Log level: Info

```
09-17 08:29:24.662   1768-1768/ecodcnc.com.helloworld I/art : Not late-enabling -Xcheck:jni (already on)
09-17 08:29:24.698   1768-1768/ecodcnc.com.helloworld I/ecodcnc.com.helloworld.MainActivity : Hello Main Activity Logging Started
09-17 08:29:24.777   1768-1768/ecodcnc.com.helloworld I/ecodcnc.com.helloworld.MainActivity : Hello Main Activity Logging Ended
09-17 08:29:24.876   1768-1787/ecodcnc.com.helloworld I/OpenGLRenderer : Initialized EGL, version 1.4
09-17 08:29:24.904   1768-1787/ecodcnc.com.helloworld W/EGL_emulation : eglSurfaceAttrib not implemented
09-17 08:29:24.904   1768-1787/ecodcnc.com.helloworld W/OpenGLRenderer : Failed to set EGL_SWAP_BEHAVIOR on surface 0xb3f181e0, error=EGL_SUCCESS
```

Terminal | 6: Android | 0: Messages | 4: Run | TODO

Session 'app': running (7 minutes ago)

Lab4 – Induce an Error

© 2015 IBM Corporation

Induce an Error in MainActivity.java - Screenshot

HelloWorld > app > src > main > java > ecodcnc > com > helloworld > MainActivity

MainActivity.java | activity_main.xml

```
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {

            // ADD Lab 3 - Challenge !!!


            // LAB 4 CODE
            int i = 0;
            int j = 0;
            int k = i / j;

            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

Emulator Nexus_6_API_22 Android 5.1 (API 22) | ecodcnc.com.helloworld (1867)

logcat | ADB logs | Memory | CPU | Log level: Info

```
09-17 09:48:13.909    1867-1882/ecodcnc.com.helloworld W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xb42a0380, error=EGL_SUCCESS
09-17 09:48:41.186    1867-1882/ecodcnc.com.helloworld W/EGL_emulation: eglSurfaceAttrib not implemented
09-17 09:48:41.186    1867-1882/ecodcnc.com.helloworld W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xb41a7720, error=EGL_SUCCESS
09-17 09:48:45.737    1867-1867/ecodcnc.com.helloworld E/AndroidRuntime: FATAL EXCEPTION: main
    Process: ecodcnc.com.helloworld, PID: 1867
    java.lang.ArithmeticException: divide by zero
            at ecodcnc.com.helloworld.MainActivity.onOptionsItemSelected(MainActivity.java:54)
            at android.app.Activity.onMenuItemSelected(Activity.java:2885)
            at com.android.internal.policy.impl.PhoneWindow.onMenuItemSelected(PhoneWindow.java:1133)
            at com.android.internal.view.menu.MenuBuilder.dispatchMenuItemSelected(MenuBuilder.java:761)
            at com.android.internal.view.menu.MenuItemImpl.invoke(MenuItemImpl.java:152)
            at com.android.internal.view.menu.MenuBuilder.performItemAction(MenuBuilder.java:904)
            at com.android.internal.view.menu.MenuBuilder.performItemAction(MenuBuilder.java:894)
```

4: Run | TODO | 6: Android | Terminal | 0: Messages

Session 'app': running (2 minutes ago) 21:82

Induce an Error in MainActivity.java

- ADD THESE LINES highlighted in green to onOptionsItemSelected method

```
int id = item.getItemId();
//noinspection SimplifiableIfStatement
if (id == R.id.action_settings) {

    // ADD Lab 3 - Challenge !!!


    // LAB 4 CODE
    int i = 0;
    int j = 0;
    int k = i / j;

    return true;
}

return super.onOptionsItemSelected(item);
```

© 2015 IBM Corporation

Run the App → Click on Settings → See the Error

Android

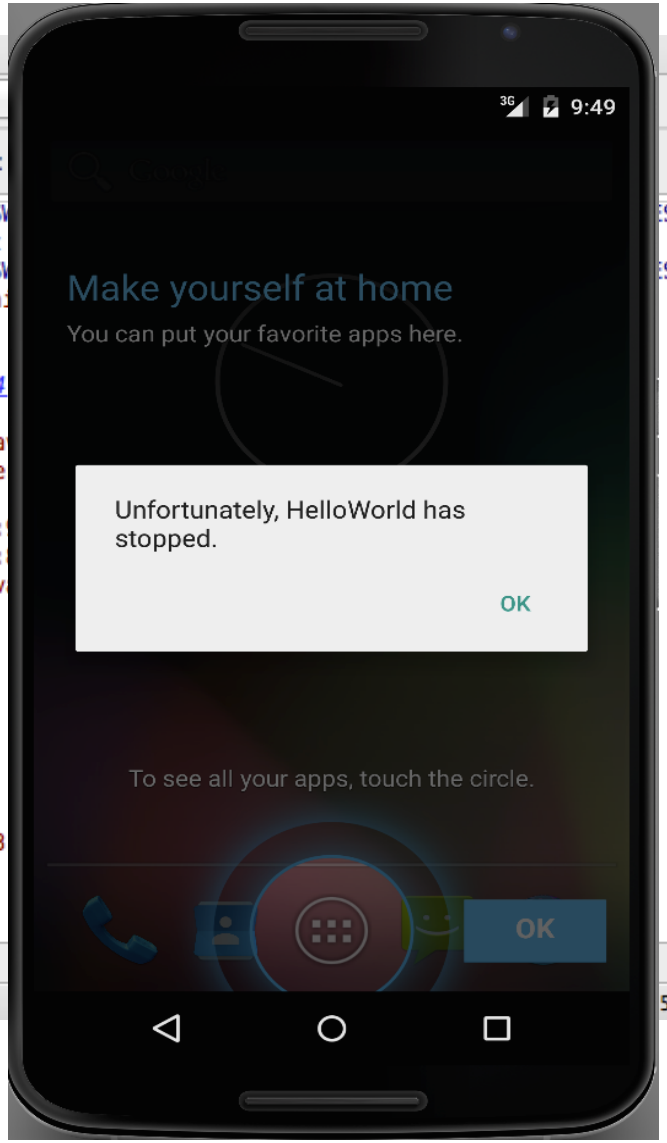
Emulator Nexus_6_API_22 Android 5.1 (API 22) | ecodcnc.com.helloworld (1867) [DEAD]

logcat | ADB logs | Memory | CPU | Log level:

```
09-17 09:48:13.909    1867-1882/ecodcnc.com.helloworld W/OpenGLRenderer: Failed to set EGL_SW
09-17 09:48:41.186    1867-1882/ecodcnc.com.helloworld W/EGL_emulation: eglSurfaceAttrib not
09-17 09:48:41.186    1867-1882/ecodcnc.com.helloworld W/OpenGLRenderer: Failed to set EGL_SW
09-17 09:48:45.737    1867-1867/ecodcnc.com.helloworld E/AndroidRuntime: FATAL EXCEPTION: ma
    Process: ecodcnc.com.helloworld, PID: 1867
    java.lang.ArithmeticException: divide by zero
            at ecodcnc.com.helloworld.MainActivity.onOptionsItemSelected(MainActivity.java:54
            at android.app.Activity.onMenuItemSelected(Activity.java:2885)
            at com.android.internal.policy.impl.PhoneWindow.onMenuItemSelected(PhoneWindow.ja
            at com.android.internal.view.menu.MenuBuilder.dispatchMenuItemSelected(MenuBuilde
            at com.android.internal.view.menu.MenuItemImpl.invoke(MenuItemImpl.java:152)
            at com.android.internal.view.menu.MenuBuilder.performItemAction(MenuBuilder.java:
            at com.android.internal.view.menu.MenuBuilder.performItemAction(MenuBuilder.java:
            at com.android.internal.view.menu.MenuPopupHelper.onItemClick(MenuPopupHelper.jav
            at android.widget.AdapterView.performItemClick(AdapterView.java:305)
            at android.widget.AbsListView.performItemClick(AbsListView.java:1146)
            at android.widget.AbsListView$PerformClick.run(AbsListView.java:3053)
            at android.widget.AbsListView$3.run(AbsListView.java:3860)
            at android.os.Handler.handleCallback(Handler.java:739)
            at android.os.Handler.dispatchMessage(Handler.java:95)
            at android.os.Looper.loop(Looper.java:135)
            at android.app.ActivityThread.main(ActivityThread.java:5257) <2 internal calls>
            at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:903
            at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:698)
09-17 09:53:45.840    1867-1867/? I/Process: Sending signal. PID: 1867 SIG: 9
```

4: Run | TODO | 6: Android | Terminal | 0: Messages

Session 'app': running (13 minutes ago)



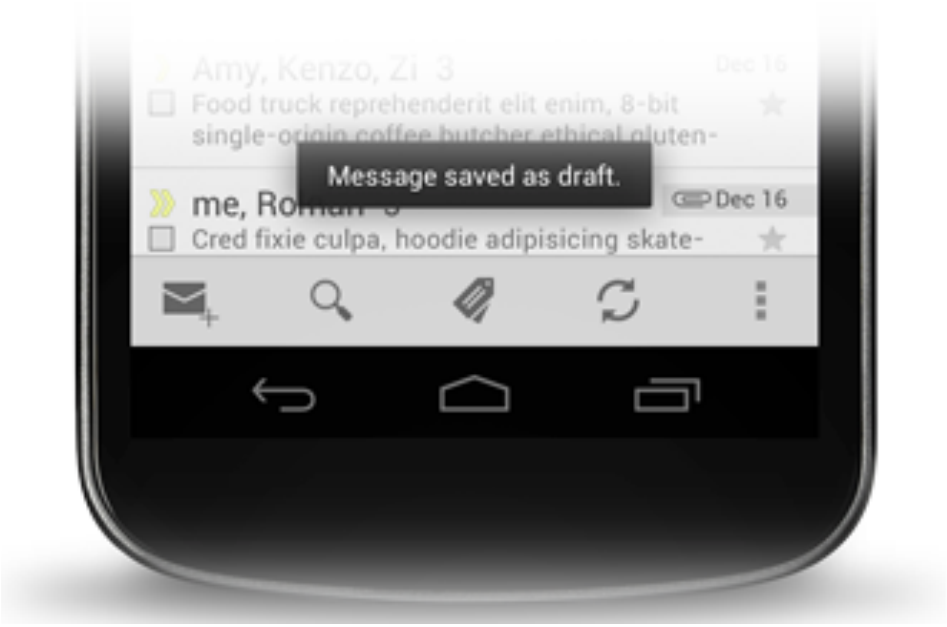
© 2015 IBM Corporation

Lab5 – Toasting

© 2015 IBM Corporation

What is a Toast?

A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive.



© 2015 IBM Corporation

Modify MainActivity.java by adding Toast

- Add to Import highlighted in green

```
import android.widget.Toast;
```

- ADD highlighted in green to onOptionsItemSelected method

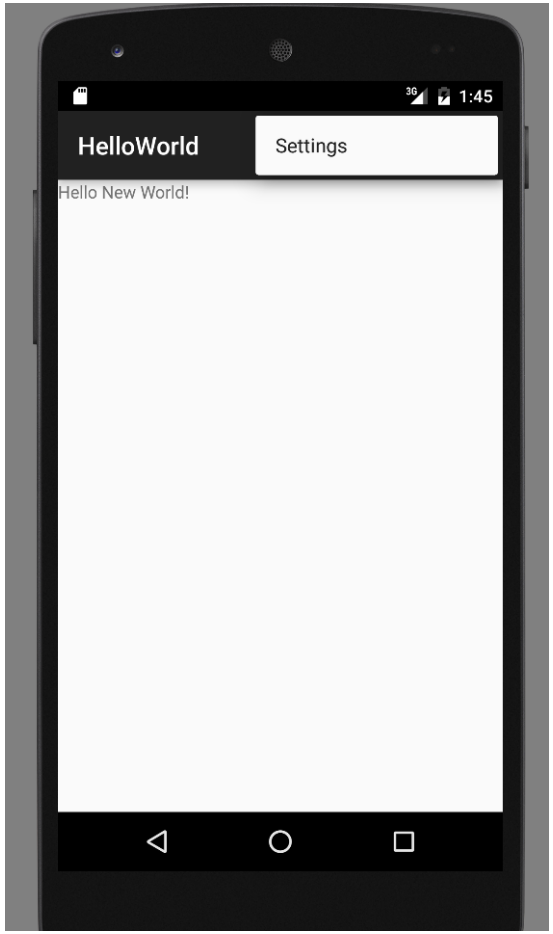
```
int id = item.getItemId();
//noinspection SimplifiableIfStatement
if (id == R.id.action_settings) {

Toast toast = Toast.makeText(this, "CLICKED SETTINGS !!!",Toast.LENGTH_LONG);
toast.show();
```

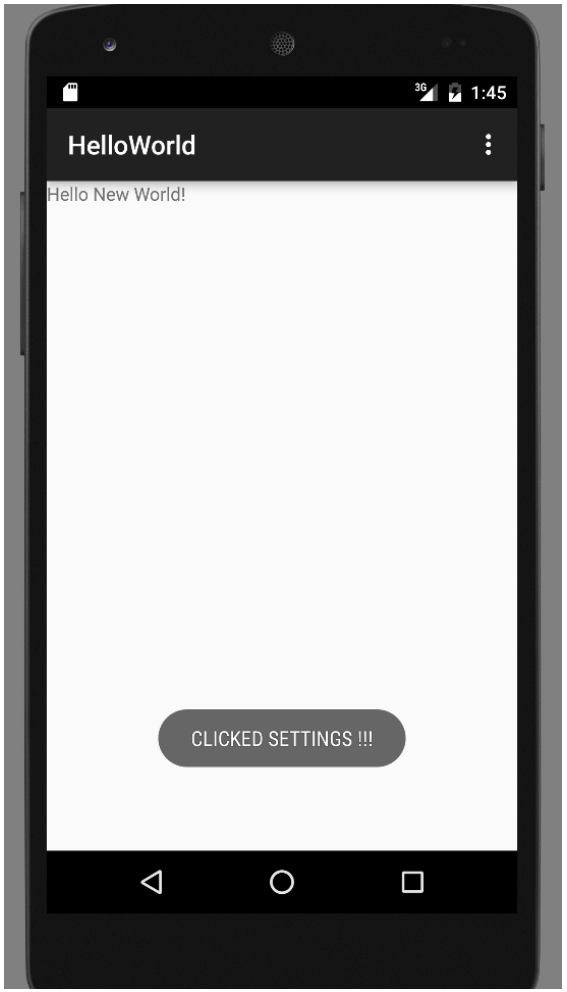
- REMOVE highlighted in green to onOptionsItemSelected method

```
// LAB 4 CODE
//int i = 0;
//int j = 0;
//int k = i / j;
```

© 2015 IBM Corporation

Run the App → View the Toast by clicking on settings (right hand corner)





© 2015 IBM Corporation

Lab6 – Debugging

© 2015 IBM Corporation

Add a Breakpoint → Run the app in Debug Mode

```
HelloWorld > app > src > main > java > ecodcnc > com > helloworld > MainActivity
MainActivity.java   activity_main.xml

        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {  item: "Settings"
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();  id: 2131230720  item: "Settings"

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {  id: 2131230720

            Toast toast = Toast.makeText(this, "CLICKED SETTINGS !!!", Toast.LENGTH_LONG);  toast: Toast@3624
            toast.show();

            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

© 2015 IBM Corporation

Debug Window

```
HelloWorld > app > src > main > java > ecodcnc > com > helloworld > MainActivity
```

```
MainActivity.java ×   activity_main.xml ×
```

```
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {  item: "Settings"
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();  id: 2131230720  item: "Settings"

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {  id: 2131230720

            Toast toast = Toast.makeText(this, "CLICKED SETTINGS !!!", Toast.LENGTH_LONG);  toast: Toast@3624
            toast.show();

            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

```
app
  manifests
  java
    ecodcnc.com.helloworld
      MainActivity
    ecodcnc.com.helloworld (androidTest)
  res
    drawable
    layout
      activity_main.xml
    menu
    mipmap
    values
Gradle Scripts
```

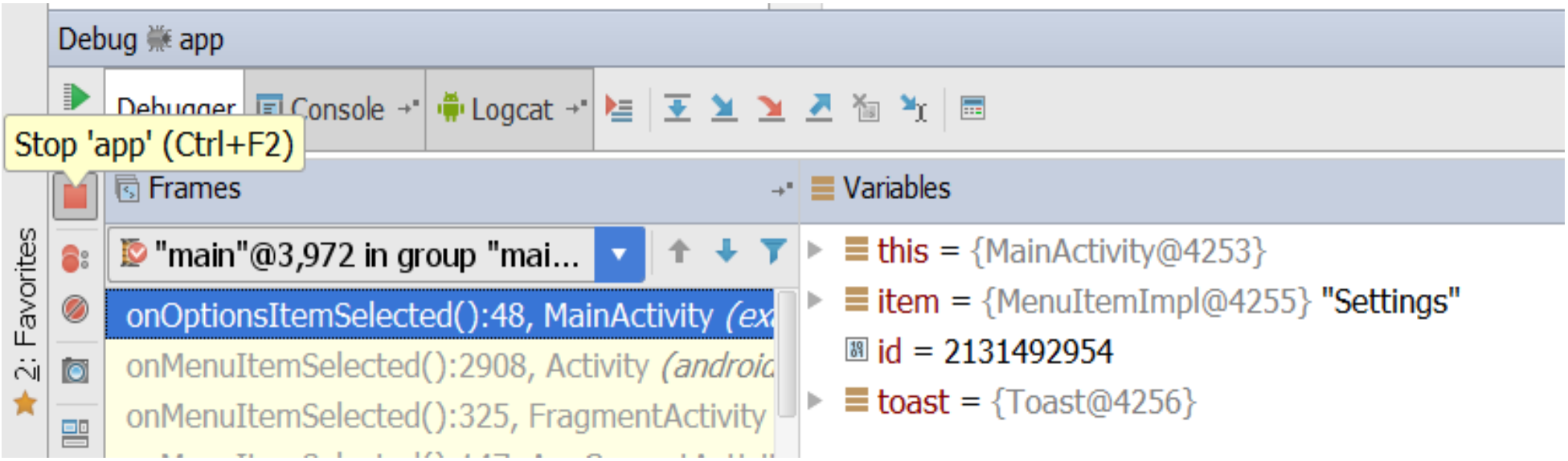
```
Debug  app
Debugger | Console | Logcat
```

```
Frames
"main"@3,589 in group "main": WAIT
onOptionsItemSelected():50, MainActivity (ecodcnc.com.helloworld)
onMenuItemSelected():2885, Activity (android.app)
onMenuItemSelected():1133, PhoneWindow (com.android.internal.policy.
dispatchMenuItemSelected():761, MenuBuilder (com.android.internal.vie
invoke():152, MenuItemImpl (com.android.internal.view.menu)
performItemAction():904, MenuBuilder (com.android.internal.view.menu
performItemAction():894, MenuBuilder (com.android.internal.view.menu
onItemClick():186, MenuPopupHelper (com.android.internal.view.menu)
performItemClick():305, AdapterView (android.widget)
```

```
Variables
this = {MainActivity@3619}
item = {MenuItemImpl@3623} "Settings"
id = 2131230720
toast = {Toast@3624}
```

```
4: Run | 5: Debug | TODO | 6: Android | Terminal | 0: Messages
Session 'app': running (2 minutes ago)   50:1  LF÷  UTF-8÷
```

Stop the Debugger



© 2015 IBM Corporation

Debugging with Android Studio

© 2015 IBM Corporation

Android Studio

HelloWorld > captures > SystemInfo_2015.09.23_19.04.01.txt

SystemInfo_2015.09.23_19.03.10.txt | SystemInfo_2015.09.23_19.03.54.txt | SystemInfo_2015.09.23_19.04.01.txt

```
Applications Memory Usage (kB):
Uptime: 761944 Realtime: 761944

** MEMINFO in pid 1828 [ecodcnc.com.helloworld] **
                   Pss  Private  Private  Swapped     Heap     Heap     Heap
                 Total    Dirty    Clean    Dirty     Size    Alloc     Free
                ------   ------   ------   ------   ------   ------   ------
  Native Heap     2518     2452        0        0    16384    12958     3425
  Dalvik Heap     1074     1028        0        0     1534      647      887
 Dalvik Other      204      204        0        0
        Stack      128      128        0        0
    Other dev        4        0        4        0
     .so mmap     1020      108      184        0
    .apk mmap      110        0        0        0
    .ttf mmap       22        0        0        0
    .dex mmap       20        0       16        0
    .oat mmap     1613        0      452        0
    .art mmap     4369     4004      136        0
   Other mmap        9        4        0        0
      Unknown       73       72        0        0
        TOTAL    11164     8000      792        0    17918    13605     4312
```

Emulator Nexus_6_API_22 Android 5.1 (API 22) | ecodcnc.com.helloworld (1828)

logcat | ADB logs | Memory | CPU | Log level: Verbose | Show only selected application

```
09-23 18:51:38.312   1828-1828/? I/art : Not late-enabling -Xcheck:jni (already on)
09-23 18:51:38.336   1828-1828/ecodcnc.com.helloworld W/ActivityThread : Application ecodcnc.com.helloworld is waiting for the debugger on port 8100...
09-23 18:51:38.340   1828-1828/ecodcnc.com.helloworld I/System.out : Sending WAIT chunk
09-23 18:51:38.359   1828-1835/ecodcnc.com.helloworld I/art : Debugger is active
09-23 18:51:38.552   1828-1828/ecodcnc.com.helloworld I/System.out : Debugger has connected
09-23 18:51:38.552   1828-1828/ecodcnc.com.helloworld I/System.out : waiting for debugger to settle...
09-23 18:51:38.763   1828-1828/ecodcnc.com.helloworld I/System.out : waiting for debugger to settle...
09-23 18:51:38.971   1828-1828/ecodcnc.com.helloworld I/System.out : waiting for debugger to settle...
09-23 18:51:39.181   1828-1828/ecodcnc.com.helloworld I/System.out : waiting for debugger to settle...
09-23 18:51:39.391   1828-1828/ecodcnc.com.helloworld I/System.out : waiting for debugger to settle...
09-23 18:51:39.602   1828-1828/ecodcnc.com.helloworld I/System.out : waiting for debugger to settle...
09-23 18:51:39.812   1828-1828/ecodcnc.com.helloworld I/System.out : waiting for debugger to settle...
09-23 18:51:40.021   1828-1828/ecodcnc.com.helloworld I/System.out : waiting for debugger to settle...
09-23 18:51:40.231   1828-1828/ecodcnc.com.helloworld I/System.out : debugger has settled (1326)
09-23 18:51:40.278   1828-1828/ecodcnc.com.helloworld I/ecodcnc.com.helloworld.MainActivity : HelloWorld Logging Started
09-23 18:54:11.317   1828-1828/ecodcnc.com.helloworld I/ecodcnc.com.helloworld.MainActivity : HelloWorld Logging Ended
09-23 18:54:56.093   1828-3639/ecodcnc.com.helloworld D/OpenGLRenderer : Use EGL_SWAP_BEHAVIOR_PRESERVED: true
09-23 18:54:56.096   1828-1828/ecodcnc.com.helloworld D/ : HostConnection::get() New Host Connection established 0xb42e9f80, tid 1828
09-23 18:54:56.100   1828-1828/ecodcnc.com.helloworld D/Atlas : Validating map...
09-23 18:54:56.158   1828-3639/ecodcnc.com.helloworld D/ : HostConnection::get() New Host Connection established 0xa5c2c180, tid 3639
09-23 18:54:56.167   1828-3639/ecodcnc.com.helloworld I/OpenGLRenderer : Initialized EGL, version 1.4
09-23 18:54:56.174   1828-3639/ecodcnc.com.helloworld D/OpenGLRenderer : Enabling debug mode 0
09-23 18:54:56.187   1828-3639/ecodcnc.com.helloworld W/EGL_emulation : eglSurfaceAttrib not implemented
09-23 18:54:56.187   1828-3639/ecodcnc.com.helloworld W/OpenGLRenderer : Failed to set EGL_SWAP_BEHAVIOR on surface 0xb439cbe0, error=EGL_SUCCESS
09-23 19:02:09.770   1828-1841/ecodcnc.com.helloworld I/art : Explicit concurrent mark sweep GC freed 7123(311KB) AllocSpace objects, 0(0B) LOS objects, 60%
  free, 659KB/1683KB, paused 1.028ms total 10.192ms
```

4: Run | 5: Debug | TODO | 6: Android | Terminal | 0: Messages | Event Log | Gradle Console

Gradle build finished in 887ms (14 minutes ago)

Lab7 – ButtonListener

© 2015 IBM Corporation

Create a New Project → Name it Lab7

Create New Project

New Project
Android Studio

Configure your new project

Application name: Lab7

Company Domain: com.ecodcnc

Package name: ecodcnc.com.lab7 Edit

Project location: /Users/bvellanki/git/javaplays-mobilestarterkit/Lab72

Cancel Previous Next Finish

© 2015 IBM Corporation

SDK for Lab7 → Select SDK → Click Next

Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

- [x] Phone and Tablet
  - Minimum SDK: API 22: Android 5.1 (Lollipop)

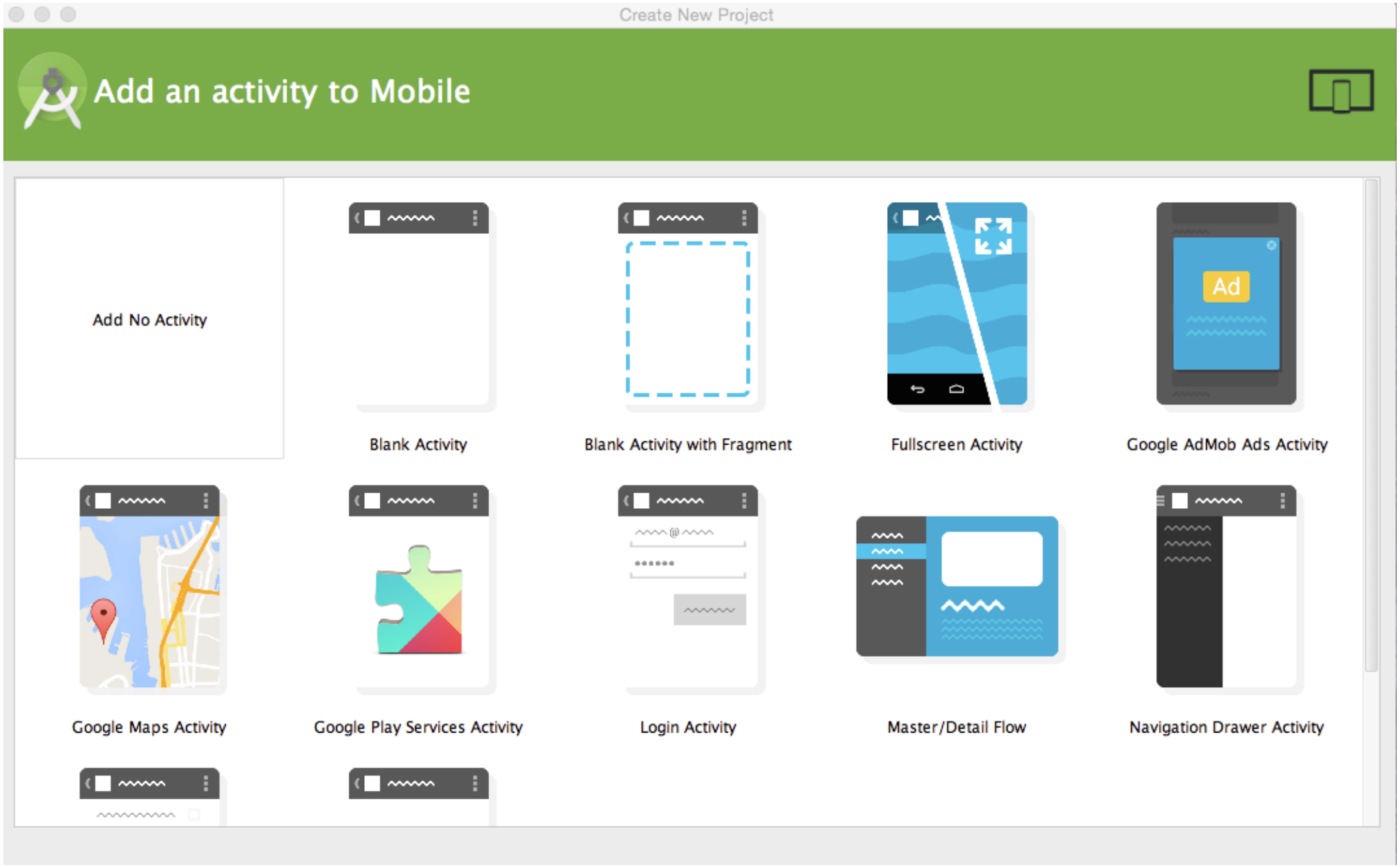
  Lower API levels target more devices, but have fewer features available. By targeting API 22 and later, your app will run on < 1% of the devices that are active on the Google Play Store.

  Help me choose
- [ ] Wear
  - Minimum SDK: API 21: Android 5.0 (Lollipop)
- [ ] TV
  - Minimum SDK: API 21: Android 5.0 (Lollipop)
- [ ] Android Auto
- [ ] Glass (Not Installed) Download
  - Minimum SDK:

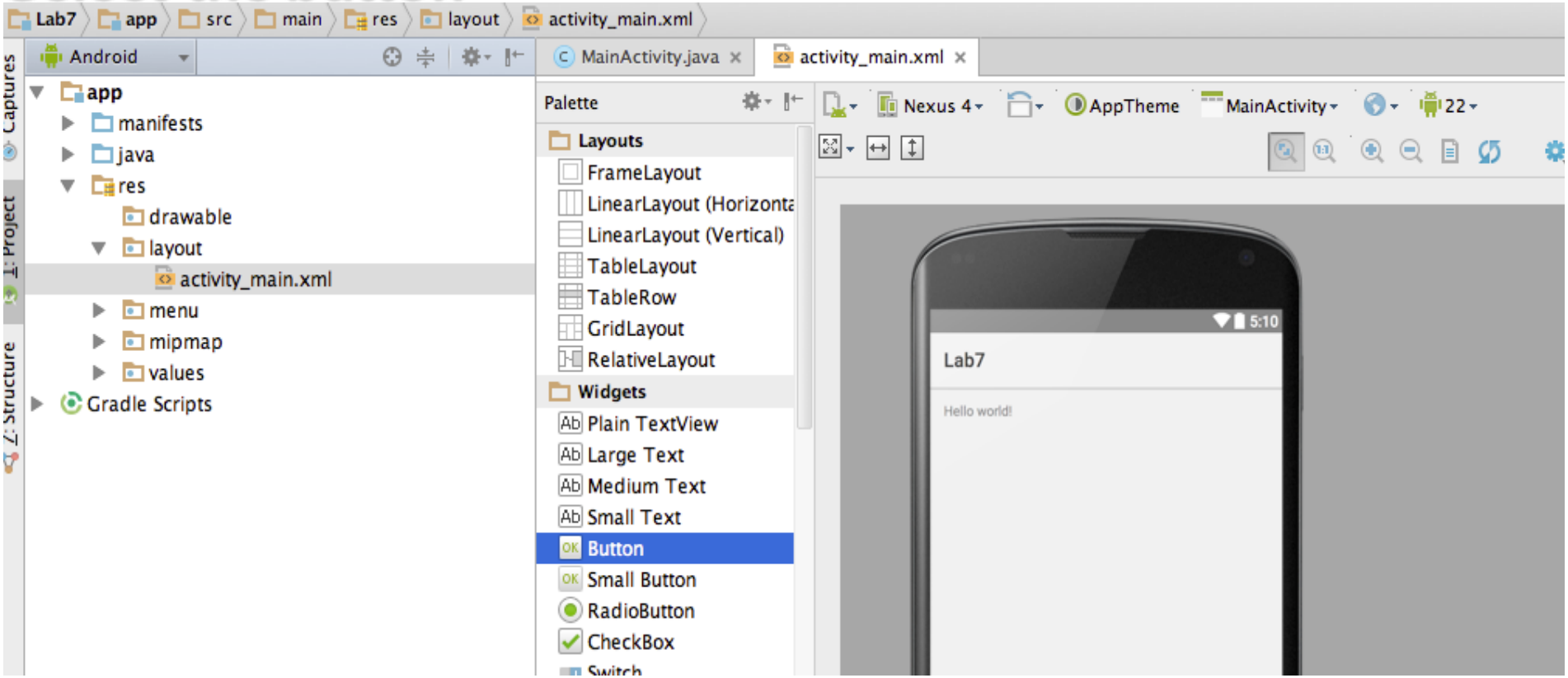
Cancel | Previous | Next | Finish

© 2015 IBM Corporation

Complete Creating Project Lab7 → Click Finish

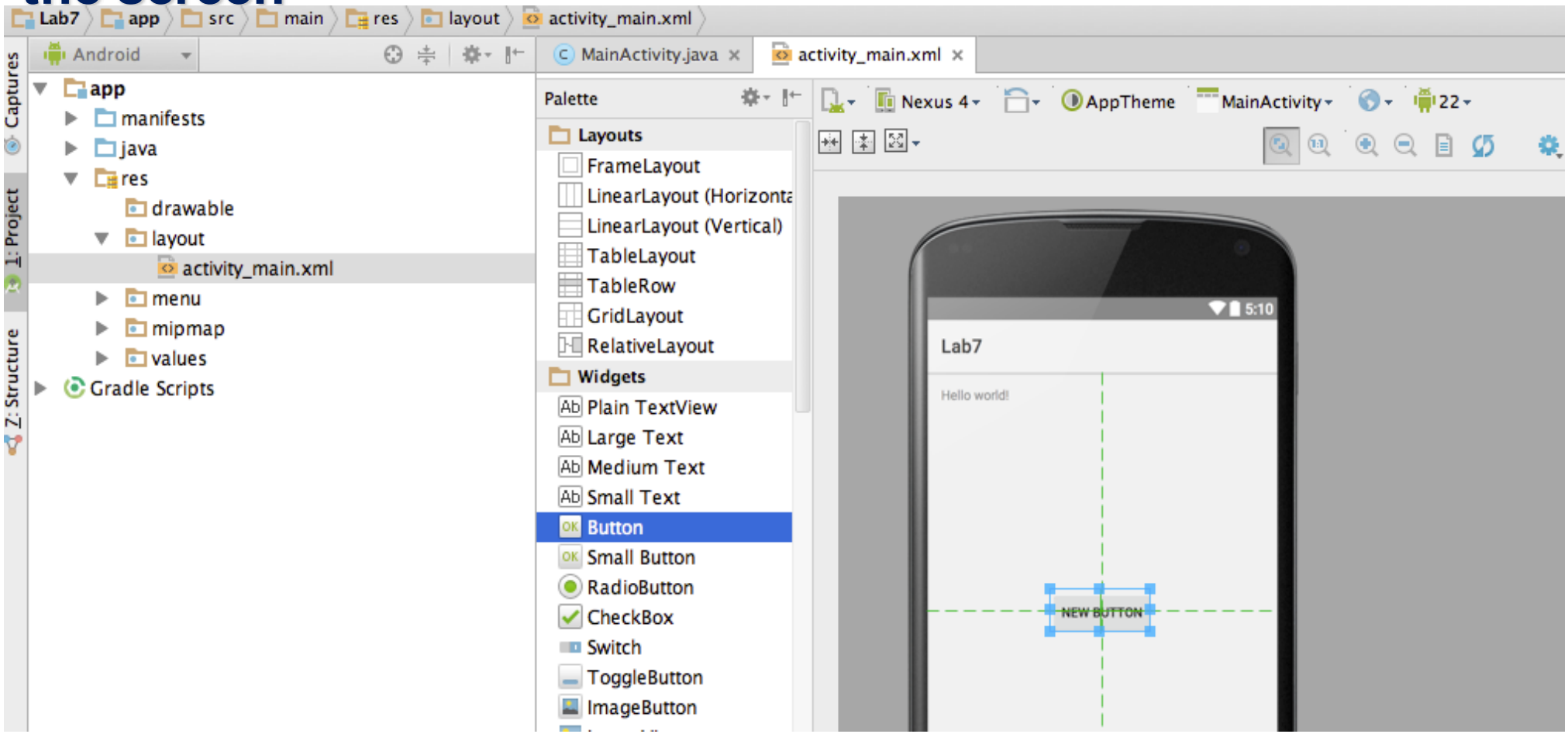


Open Activity_Main.xml → Add a Button to the screen → Select the button



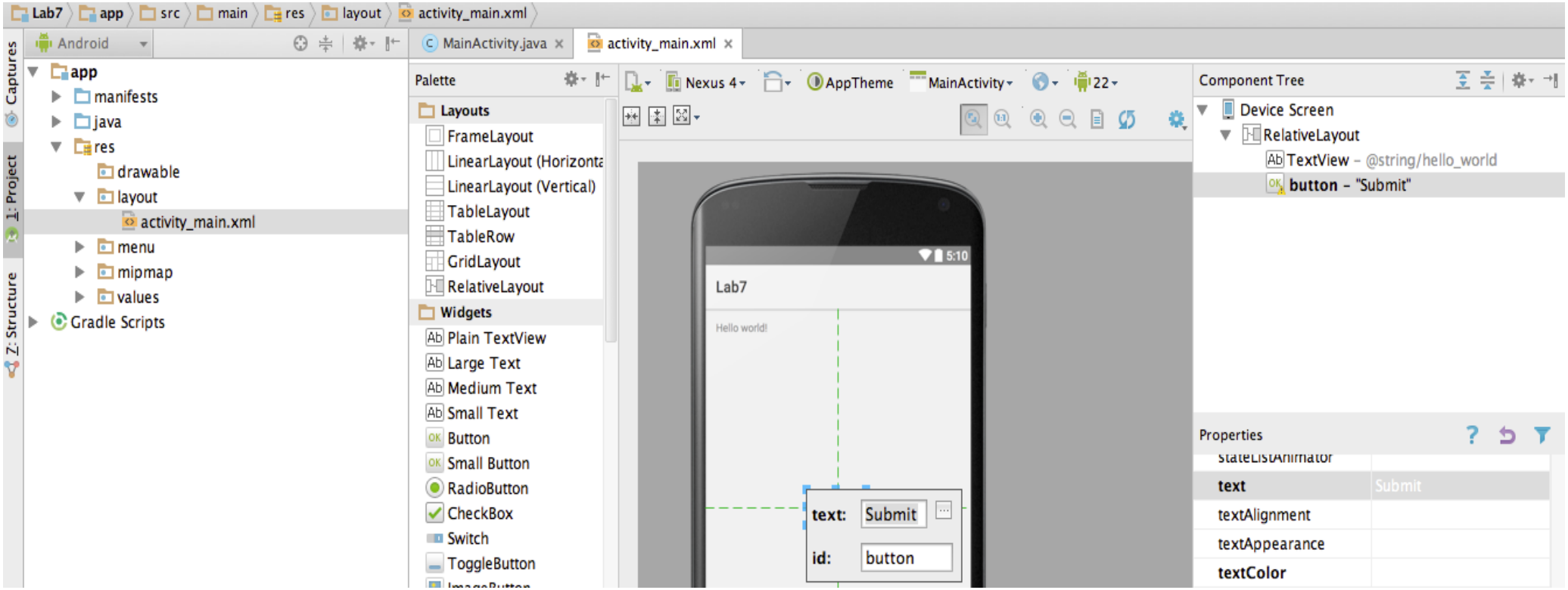
© 2015 IBM Corporation

Activity_Main.xml → Add a Button → Drag the button to the screen



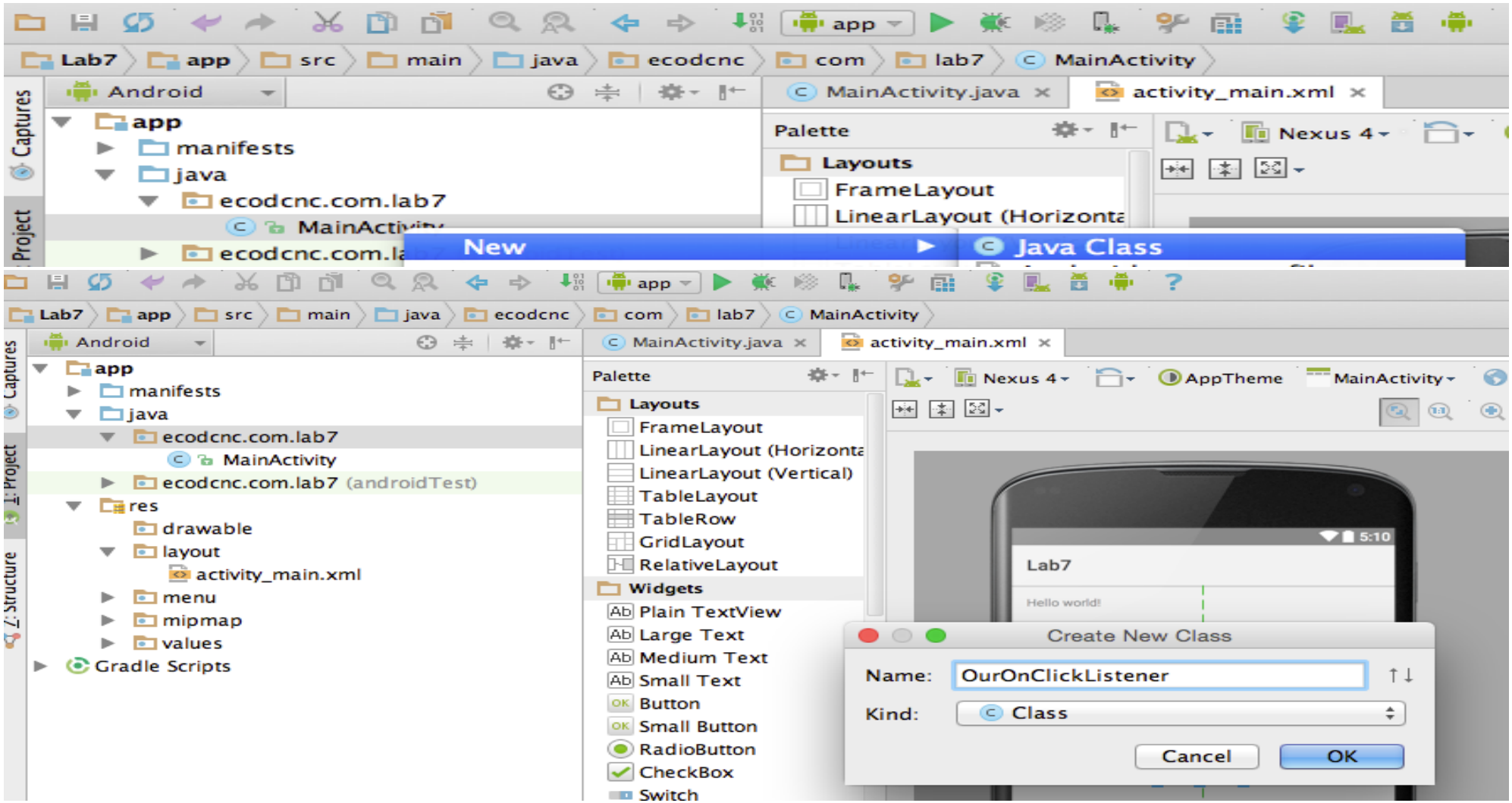
© 2015 IBM Corporation

Activity_Main.xml → Add a Button → Double Click on the Button and Name It → Submit



© 2015 IBM Corporation

Add a Listener to Button → Create a Java Class → OurOnClickListener.java → This class will listen to button event



© 2015 IBM Corporation

Add onClick Method in OurOnClickListener.java → To Listen Submit Action in Mobile UI → Screenshot

Lab7 > app > src > main > java > ecodcnc > com > lab7 > OurOnClickListener

Android

- app
  - manifests
    - AndroidManifest.xml
  - java
    - ecodcnc.com.lab7
      - MainActivity
      - OurOnClickListener
    - ecodcnc.com.lab7 (androidTest)
  - res
    - drawable
    - layout
      - activity_main.xml
    - menu
      - menu_main.xml
    - mipmap
      - ic_launcher.png (4)
    - values
      - dimens.xml (2)
      - strings.xml
      - styles.xml (2)

OurOnClickListener.java

```java
package ecodcnc.com.lab7;

import android.view.View;
import android.view.View.OnClickListener;

public class OurOnClickListener implements OnClickListener {

    MainActivity caller;
    private int count;

    public OurOnClickListener(MainActivity activity) {
        this.caller = activity;
        this.count = 0;
    }

    @Override
    public void onClick(View v) {
        count++;
        String countstr = Integer.toString(count);
        caller.textView.setText("Clicked " + countstr + " times");

    }

}
```

© 2015 IBM Corporation

Add onClick Method in OurOnClickListener.java → To Listen Submit Action in Mobile UI

Note: There will be an Error, but will be fixed in the next step

```
package ecodcnc.com.lab7;

    import android.view.View;
    import android.view.View.OnClickListener;

    public class OurOnClickListener implements OnClickListener {
        MainActivity caller;
        private int count;
    public OurOnClickListener(MainActivity activity) {
            this.caller = activity;
            this.count = 0;
        }
        @Override
        public void onClick(View v) {
            count++;
            String countstr = Integer.toString(count);
            caller.textView.setText("Clicked " + countstr + " times");
        }
    }
```

© 2015 IBM Corporation

Add onCreate Method in MainActivity.java → For Main Screen Submit Action Invocation (OurOnClickListener.java) → Screenshot

Lab7 > app > src > main > java > ecodcnc > com > lab7 > MainActivity

Android

- app
  - manifests
    - AndroidManifest.xml
  - java
    - ecodcnc.com.lab7
      - MainActivity
      - OurOnClickListener
    - ecodcnc.com.lab7 (androidTest)
  - res
    - drawable
    - layout
      - activity_main.xml
    - menu
      - menu_main.xml
    - mipmap
      - ic_launcher.png (4)
    - values
      - dimens.xml (2)
      - strings.xml
      - styles.xml (2)

MainActivity.java | OurOnClickListener.java

```java
package ecodcnc.com.lab7;

import android.view.View;
import android.view.View.OnClickListener;

public class OurOnClickListener implements OnClickListener {

    MainActivity caller;
    private int count;

    public OurOnClickListener(MainActivity activity) {
        this.caller = activity;
        this.count = 0;
    }

    @Override
    public void onClick(View v) {
        count++;
        String countstr = Integer.toString(count);
        caller.textView.setText("Clicked " + countstr + " times");

    }

}
```

Add onCreate Method in MainActivity.java → For Main Screen Submit Action Invocation (OurOnClickListener.java)

```
package  ecodcnc.com.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    TextView textView;
    Button ourButton;
    // OVER WRITE OnCreate Method in LAB 7
    @Override
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView(R.layout.activity_main);
        textView = (TextView) findViewById(R.id.textView);
        ourButton = (Button) findViewById(R.id.button);
        ourButton.setOnClickListener(new OurOnClickListener(this));
    }
```

© 2015 IBM Corporation

Run the Application

app

Lab7 > app > src > main > java > ecodcnc > com > lab7 > MainActivity

Android

- app
  - manifests
    - AndroidManifest.xml
  - java
    - ecodcnc.com.lab7
      - MainActivity
      - OurOnClickListener
    - ecodcnc.com.lab7 (androidTest)
  - res
    - drawable
    - layout
      - activity_main.xml
    - menu
      - menu_main.xml
    - mipmap
      - ic_launcher.png (4)
    - values
      - dimens.xml (2)
      - strings.xml

MainActivity.java | OurOnClickListener.java

```java
package ecodcnc.com.lab7;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.TextView;


public class MainActivity extends Activity {

    TextView textView;
    Button ourButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView = (TextView) findViewById(R.id.textView);
        ourButton = (Button) findViewById(R.id.button);
        ourButton.setOnClickListener(new OurOnClickListener(this));

    }
```

© 2015 IBM Corporation

Hurray!!! My Submit Count App is Cool → Click on the Submit button and see the count increase

© 2015 IBM Corporation

Important Files

1. src/**MainActivity.java**
    1. Activity which is started when app executes
2. gen/**R.java** (DO NOT MODIFY!)
    1. Auto-generated, auto-updated file with identifiers from main.xml, strings.xml, and elsewhere
3. res/layout/**activity_main.xml**
    1. Defines & lays out widgets for the activity
4. res/values/**strings.xml**
    1. String constants used by app
5. **AndroidManifest.xml**
    1. Declares all the app's components
    2. Names libraries app needs to be linked against
    3. Identifies permissions the app expects to be granted

© 2015 IBM Corporation

Lab7 – Appendix

© 2015 IBM Corporation

1 - src/**MainActivity.java**

```
package com.example.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

© 2015 IBM Corporation

2 - gen/**R.java**

- Auto-generated file with identifiers from main.xml, strings.xml, and elsewhere
- */* AUTO-GENERATED FILE. DO NOT MODIFY.*

```
*
* This class was automatically generated by the
* aapt tool from the resource data it found. It
* should not be modified by hand.
*/

package com.example.helloworld;

public final class R {
public static final class anim {
public static final int abc_fade_in=0x7f050000;
public static final int abc_fade_out=0x7f050001;
public static final int abc_grow_fade_in_from_bottom=0x7f050002;
public static final int abc_popup_enter=0x7f050003;
public static final int abc_popup_exit=0x7f050004;
public static final int abc_shrink_fade_out_from_bottom=0x7f050005;
public static final int abc_slide_in_bottom=0x7f050006;
public static final int abc_slide_in_top=0x7f050007;
public static final int abc_slide_out_bottom=0x7f050008;
public static final int abc_slide_out_top=0x7f050009;
}
public static final class attr {
/** <p>Must be a reference to another resource, in the form
"<code>@[+][<i>package</i>:]<i>type</i>:<i>name</i></code>"
or to a theme attribute in the form "<code>?[<i>package</i>:][<i>type</i>:]<i>name</i></code>".
```

Do not modify!

© 2015 IBM Corporation

3 - res/layout/activity_main.xml

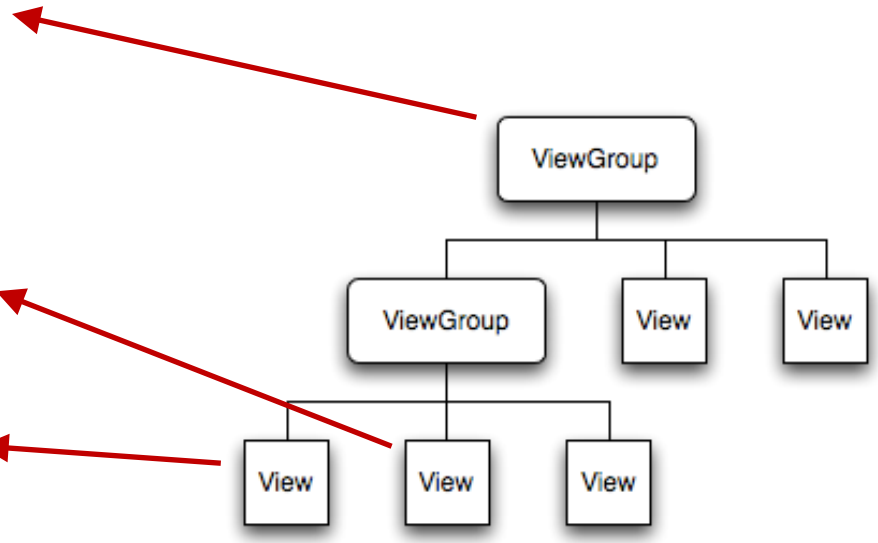
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
   android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">

   <TextView android:text="@string/hello_world" android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/textView" />

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Submit"
      android:id="@+id/button"
      android:layout_below="@+id/textView"
      android:layout_toRightOf="@+id/textView"
      android:layout_toEndOf="@+id/textView"
      android:layout_marginTop="68dp" />

</RelativeLayout>
```

© 2015 IBM Corporation

3 contd. - Understanding Layout

- Declares layouts & widgets for the activity

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <EditText
        android:id="@+id/name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <Button
        android:id="@+id/hello_button"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="Press Me" />

</LinearLayout>
```



Tree from: http://developer.android.com/guide/topics/ui/inex.html

© 2015 IBM Corporation

3 Contd - Available Layouts

LinearLayout

RelativeLayout

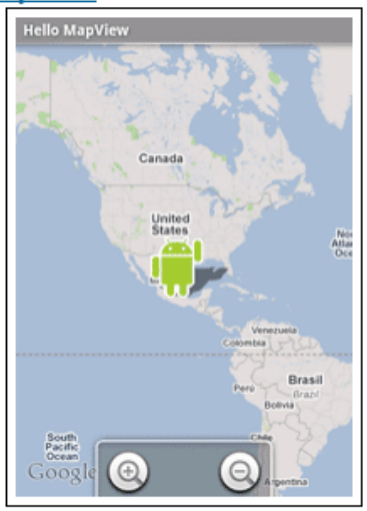
TableLayout

http://developer.android.com/resources/tutorials/views/index.html

© 2015 IBM Corporation

3 Layout and Available Widgets

MapView



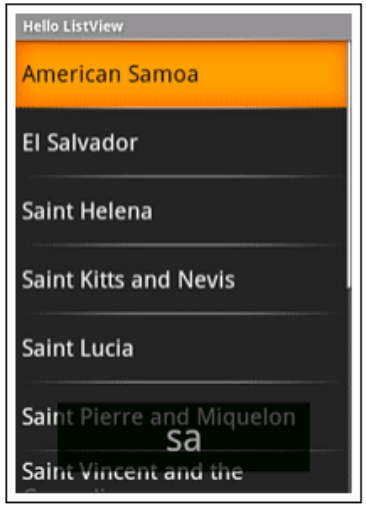
WebView

DatePicker

Spinner

AutoComplete

ListView



http://developer.android.com/resources/tutorials/views/index.html

© 2015 IBM Corporation

4 - res/values/**strings.xml**

- String constants used by app

```
<resources>
<string name="app_name">HelloWorld</string>

<string name="hello_world">Hello world!</string>
<string name="action_settings">Settings</string>
</resources>
```

- Used for supporting Localization
  - res/values-**es**/values/strings.xml to support Spanish
  - res/values-**fr**/values/strings.xml to support French
  - Etc.

© 2015 IBM Corporation

5 - AndroidManifest.xml

- Declares all the app's components
- Names libraries app needs to be linked against
- Identifies permissions the app expects to be granted

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.helloworld" >

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme" >
<activity
android:name=".MainActivity"
android:label="@string/app_name" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>
```

© IBM Corporation

5 contd - AndroidManifest.xml

All Activities that are part of application must be registered in Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="scott.examples.lifeCycleTest"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="10" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".LifeCycleTestActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
                <activity
            android:name=".NameGetter"
            android:label="@string/getName"/>
    </application>
```

Specify Activity to start with

© IBM Corporation

Some other topics covered

- onCreate() in Activity
- Adding Text and Changing Properties from Code
- Use the log to v, d, i, w, e
  VERBOSE, DEBUG, INFO, WARN, ERROR
- Create a TAG so we can filter
- Toast a message
- Debug BreakPoint
- ScreenShot of the APP
- Adding component
- Adding an Event and Setting Properties

© 2015 IBM Corporation

Understanding the Essence of Lifecycle

Necessary to overload callback methods so you app behaves well:

- App should not crash if the user receives a phone call or switches to another app while using your app.
- App should not consume valuable system resources when the user is not actively using it.
- App should not lose the user's progress if they leave your app and return to it at a later time.
- App should not crash or lose the user's progress when the screen rotates between landscape and portrait orientation.

http://developer.android.com/training/basics/activity-lifecycle/starting.html

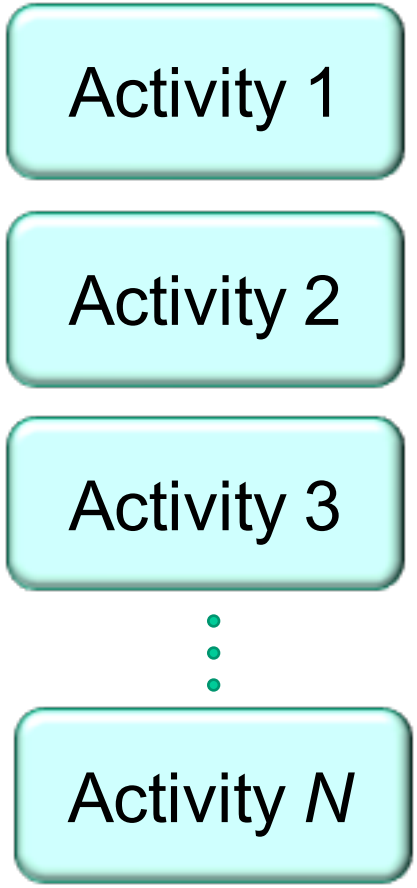
© 2015 IBM Corporation

Understanding Activity Stack

© 2015 IBM Corporation

Activity Stack

Most recently created is at Top



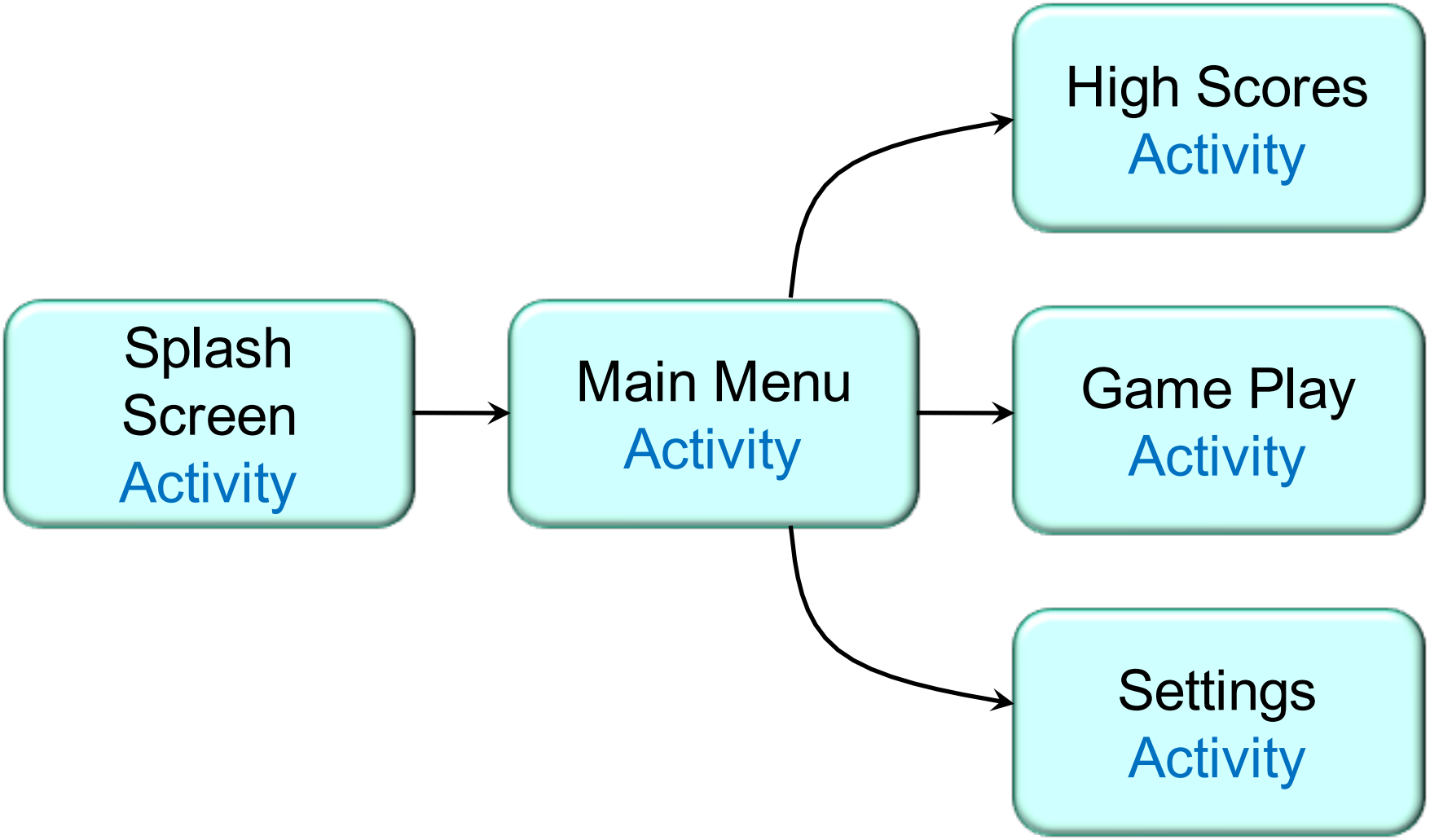
User currently interacting with me

Pressing Back or destroying A1 will bring me to the top

If Activities above me use too many resources, I'll be destroyed!

© 2015 IBM Corporation

Typical Game



Conder & Darcey (2010), Fig 4.1, p. 74

© 2015 IBM Corporation

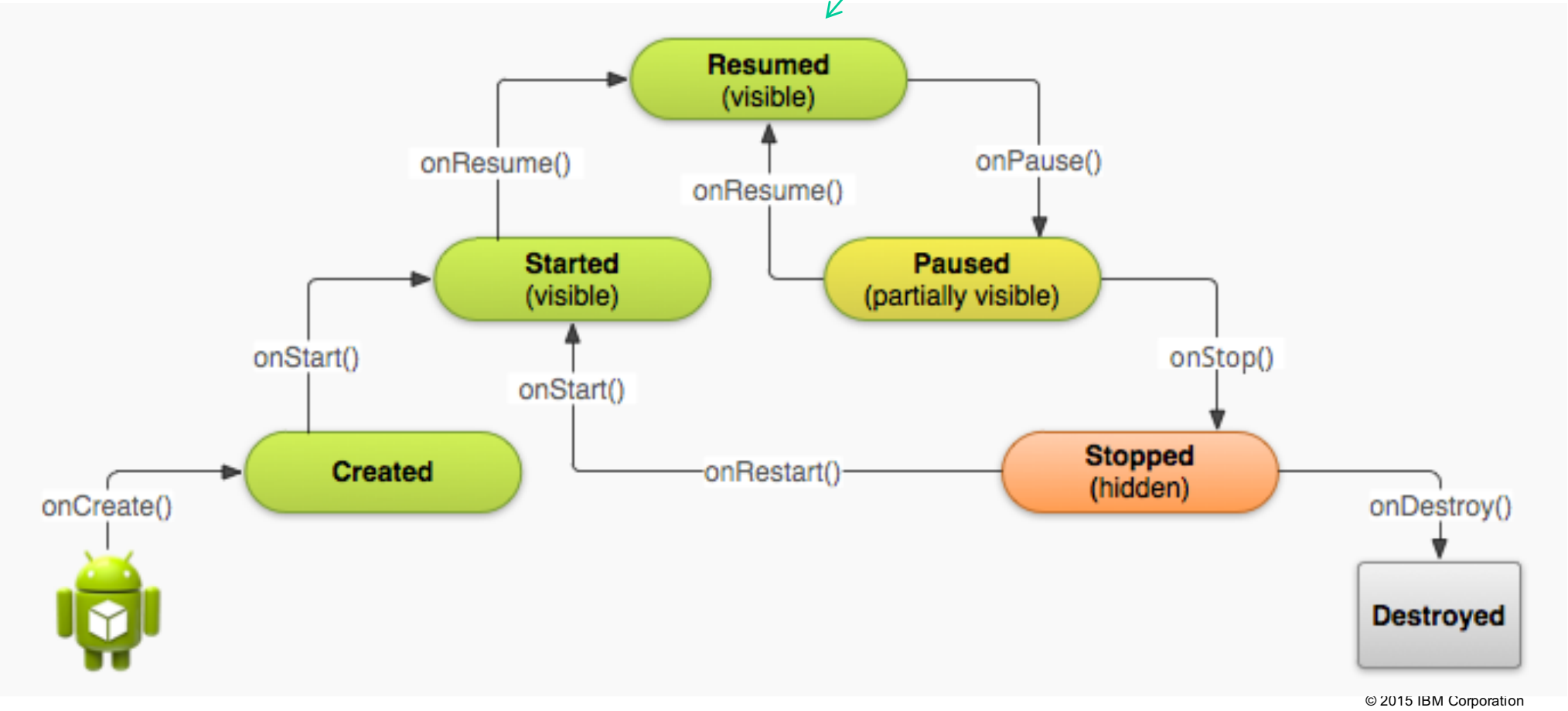
Starting Activities

- Android applications don't start with a call to main(String[])
- instead a series of callback methods are invoked
- each corresponds to specific stage of the Activity / application lifecycle
- callback methods also used to tear down Activity / application

© 2015 IBM Corporation

Simplified Lifecycle Diagram

ready to interact with user



© 2015 IBM Corporation

Primary States

- Active
  - activity is in the foreground and user can interact with it
- Paused
  - activity partially obscured by another activity and user cannot interact with it (for example when working with a menu or dialog)
- Stopped
  - activity completely hidden and not visible to user. It is in the background.
  - Activity instance and variables are retained but no code is being executed by the activity
- Dead, activity terminated (or never started)
- Two other states, Created and Started, but they are transitory onCreate -> onStart -> onResume

© 2015 IBM Corporation

Application Components List

five primary components - different purposes

Activity

- single screen with a user interface, app may have several activities, subclass of Activity
- Most of early examples will be activities

Intents

- used to pass information between applications

Service

- Application component that performs long-running operations in background with no UI
- example, an application that automatically responds to texts when driving

© 2015 IBM Corporation

References

- Android Dev Guide
  http://developer.android.com/guide/topics/fundamentals.html

- http://developer.android.com/guide/topics/fundamentals/activities.html

© 2015 IBM Corporation

Content Providers

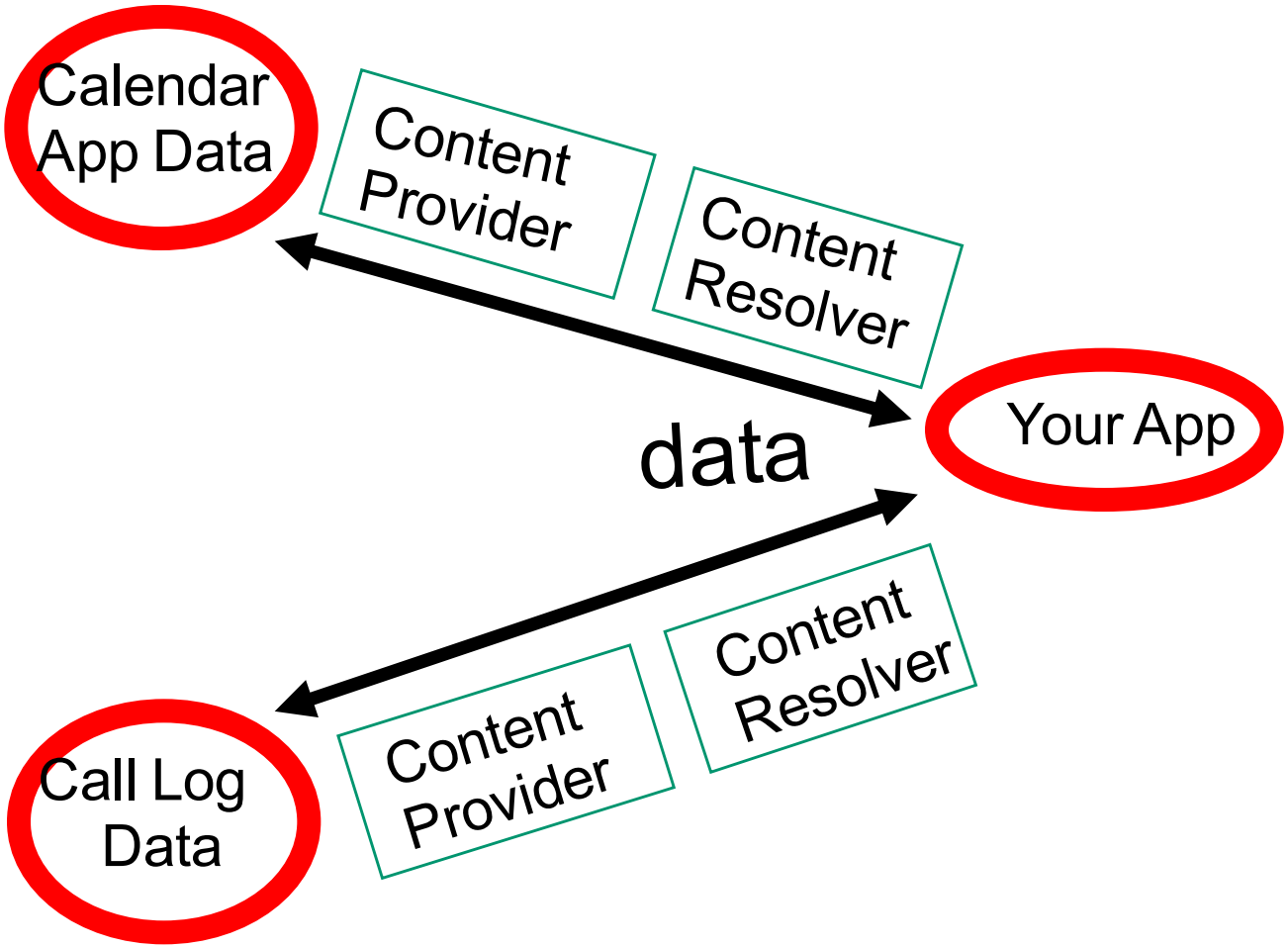
© 2015 IBM Corporation

Content Providers

- Standard mechanism / interface to allow code in one process (app, content resolver) to access data from another process (app, content provider)
  - example, app to remind you to call certain people
  - content resolver accesses call log to check last contact
- manage access to a structured set of data
- encapsulate the data and provide mechanisms for defining data security

© 2015 IBM Corporation

Content Provider - Content Resolver



© 2015 IBM Corporation

Content Providers

- Many of the built in applications have content providers to allow other apps to access data

- Examples of built in content providers
  - AlarmClock
  - CalendarContract (API level 14)
  - CallLog (sent and received calls)
  - ContactsContract
  - MediaStore (audio / visual data)
  - UserDictionary
  - VoicemailContract
  - many, many more

http://developer.android.com/reference/android/provider/package-summary.html

© 2015 IBM Corporation

Content Providers

- Provide access to a central data repository
  - ability to read and write to centralized data
- data presented by Content Provider in the form of a table
  - like table from relational database
- Each row in data table one "piece" of data in repository

© 2015 IBM Corporation

Accessing Data

- Use a ContentResolver client object in your app
- ContentResolver communicates with ContentProvider
  - provides "CRUD" functionality, **C**reate, **R**etrieve, **U**pdate, **D**elete
- matching methods in Content Resolver / Content Provider
- example: query() method
- Create a cursor via content resolver to move through rows of table

© 2015 IBM Corporation

Query

- 5 parameters
- uri for content, URI
  - look at class documentation, generally a constant
- projection, String[]
  - what columns to get from the table, null = all, inefficient
- selection clause, String
  - filter, what rows to include, a SQL WHERE clause
- selection args, String[]
  - replace ?'s in selection with these
- sortOrder, String
  - how to order the rows

© 2015 IBM Corporation

Cursor

- The ContentResolver query method creates and returns a Cursor
- Similar to a Database Cursor
  - similar to Scanner or Iterator
- Move through the data (rows) one element at a time
- Process data with loop or bind cursor to a ListView with an Adapter

© 2015 IBM Corporation

Using Selection Criteria

- Example gets rows in table
- the selection criteria and selection args allow picking only certain rows
- essentially an SQL WHERE clause
  - http://www.w3schools.com/sql/sql_where.asp
- specify a criteria that must be met
- ? is value filled in with selectionArgs
  - multiple criteria possible, AND, OR, NOT

© 2015 IBM Corporation

Permissions

- Using certain content providers require an application request permission
  - so user aware what content the application will access and possibly modify

String	READ_CALENDAR	Allows an application to read the user's calendar data.
String	READ_CALL_LOG	Allows an application to read the user's call log.
String	READ_CONTACTS	Allows an application to read the user's contacts data.

String	WRITE_CALENDAR	Allows an application to write (but not read) the user's calendar data.
String	WRITE_CALL_LOG	Allows an application to write (but not read) the user's contacts data.
String	WRITE_CONTACTS	Allows an application to write (but not read) the user's contacts data.

© 2015 IBM Corporation

Contracts

- Some apps that have content providers provide *Contract classes*
- "help work with Content provider Uris, column names, intent actions, and other features of the content provider”
- Adds a layer of abstraction and encapsulation

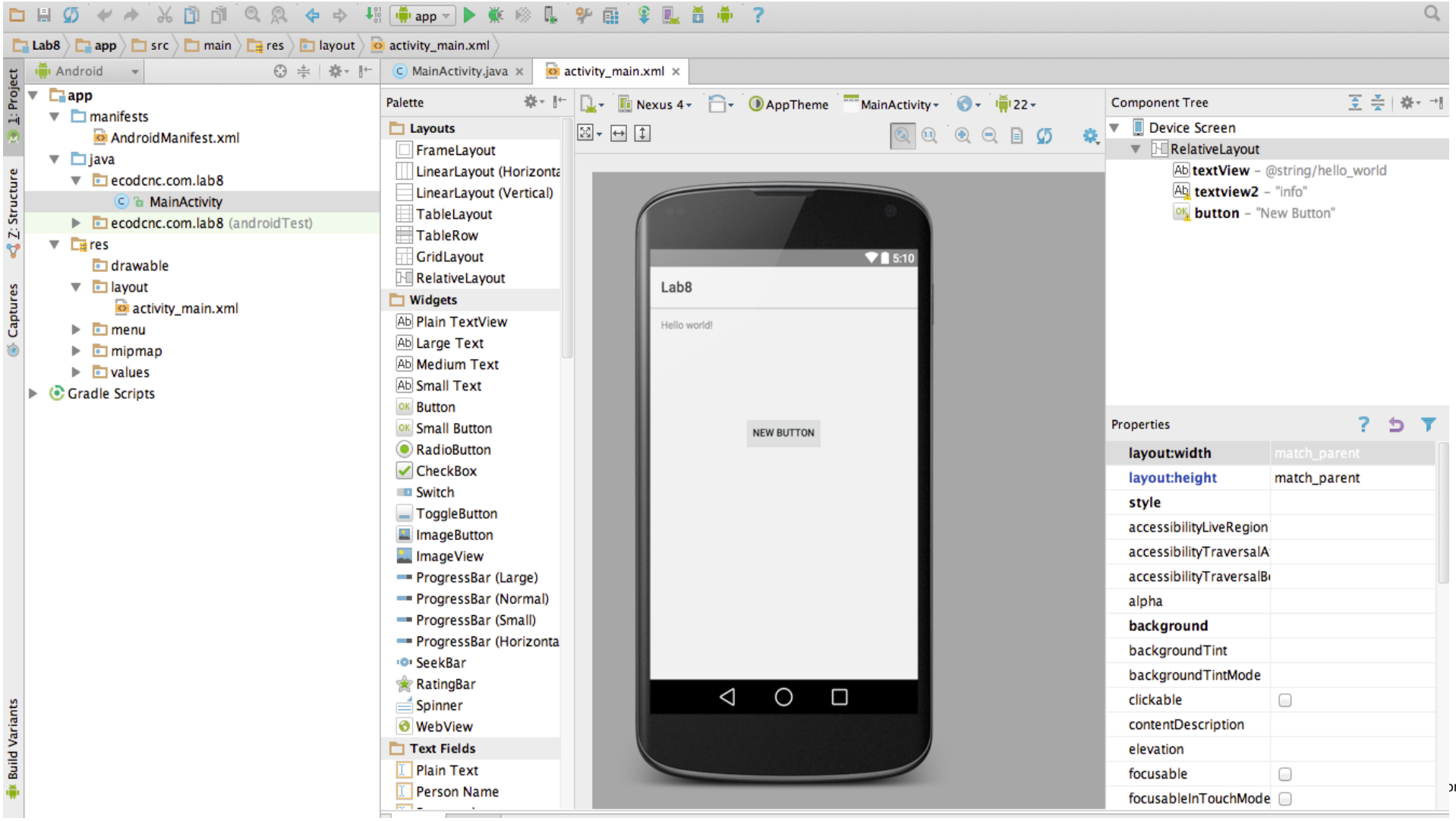
© 2015 IBM Corporation

Lab8 – Content Providers

© 2015 IBM Corporation

Create a New Project → Lab8 (Similar To Lab7) →Update activity_main.xml to build the screen layout

- Copy the code from → https://github.com/ibmecod/mobile-androidintro/blob/master/Lab8/**activity_main.xml**→ activity_main.xml



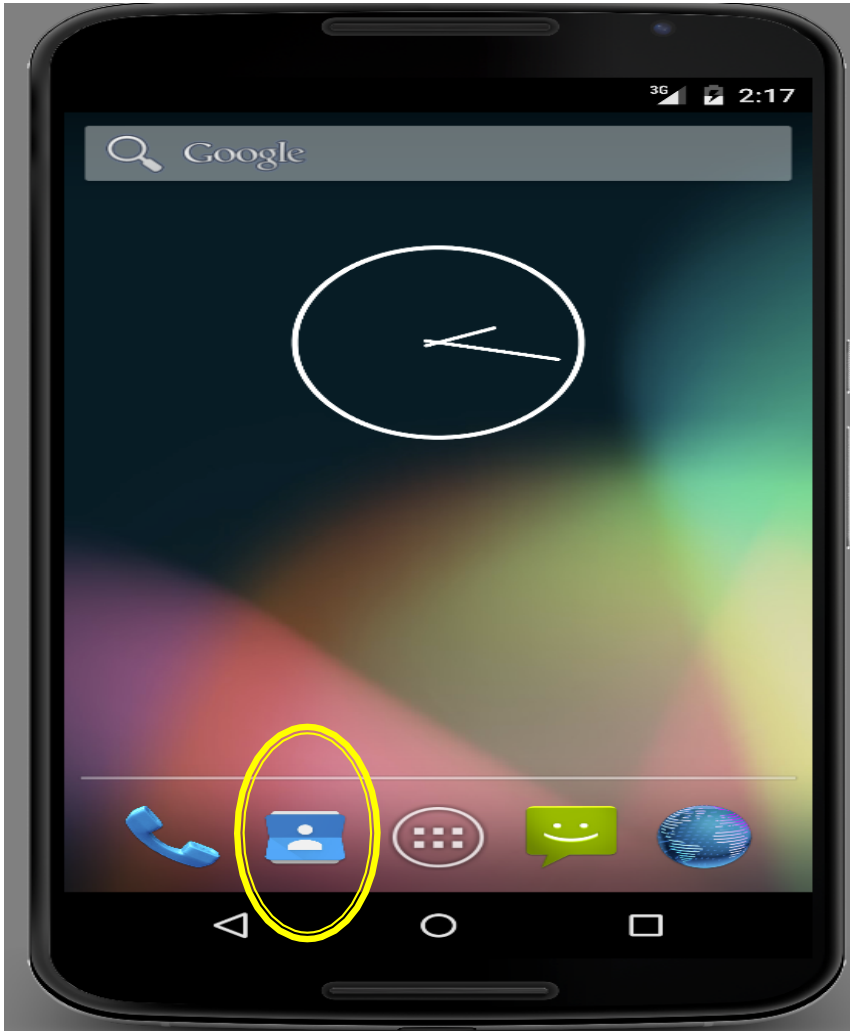
Update MainActivity.java to invoke content providers

- Copy the code from → https://github.com/ibmecod/mobile-androidintro/blob/master/Lab8/MainActivity.java → MainActivity.java

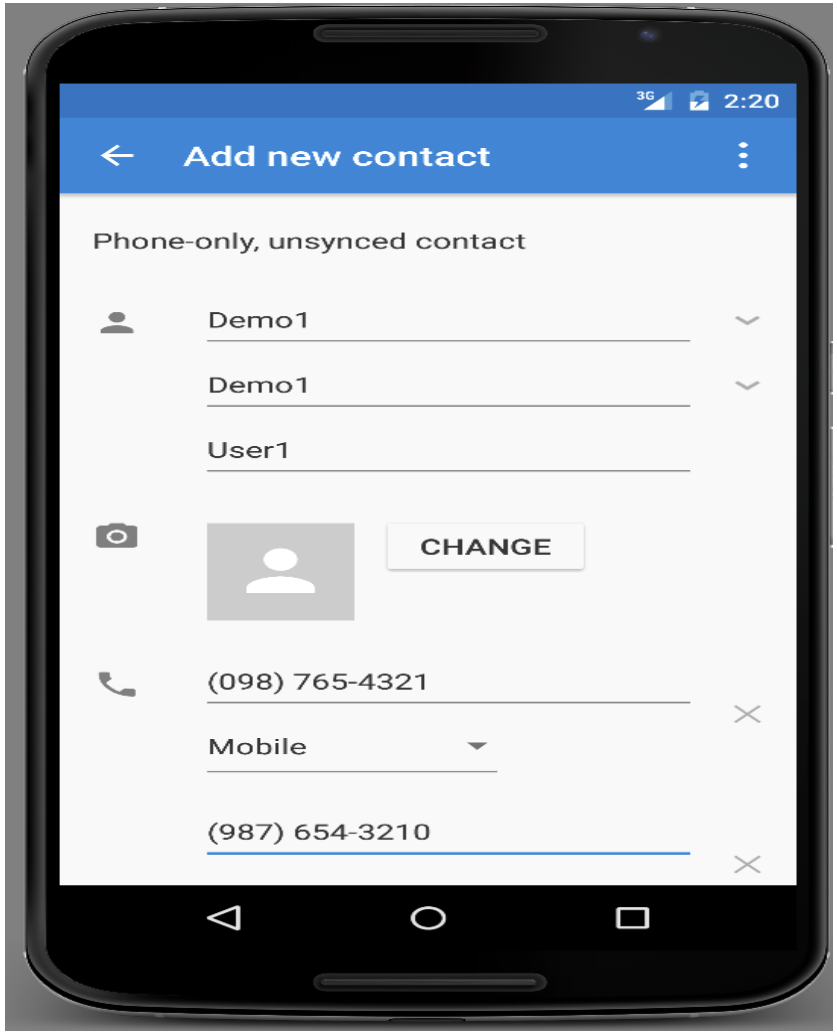
```
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void MyClick(View view) {

        Log.d("MyClick", "Yes working");

        TextView myview = (TextView) findViewById(R.id.textview2);
        myview.setVisibility(View.VISIBLE);

        myview.setText(getContacts());


    }

    // new for content provider
    private String getContacts() {
        String phoneNumber = null;
        String email = null;

        Uri CONTENT_URI = ContactsContract.Contacts.CONTENT_URI;
        String _ID = ContactsContract.Contacts._ID;
        String DISPLAY_NAME = ContactsContract.Contacts.DISPLAY_NAME;
        String HAS_PHONE_NUMBER = ContactsContract.Contacts.HAS_PHONE_NUMBER;

        Uri PhoneCONTENT_URI = ContactsContract.CommonDataKinds.Phone.CONTENT_URI;
        String Phone_CONTACT_ID = ContactsContract.CommonDataKinds.Phone.CONTACT_ID;
        String NUMBER = ContactsContract.CommonDataKinds.Phone.NUMBER;
```

Run the Application → Create Contacts

- Click on Contacts → Add Contacts



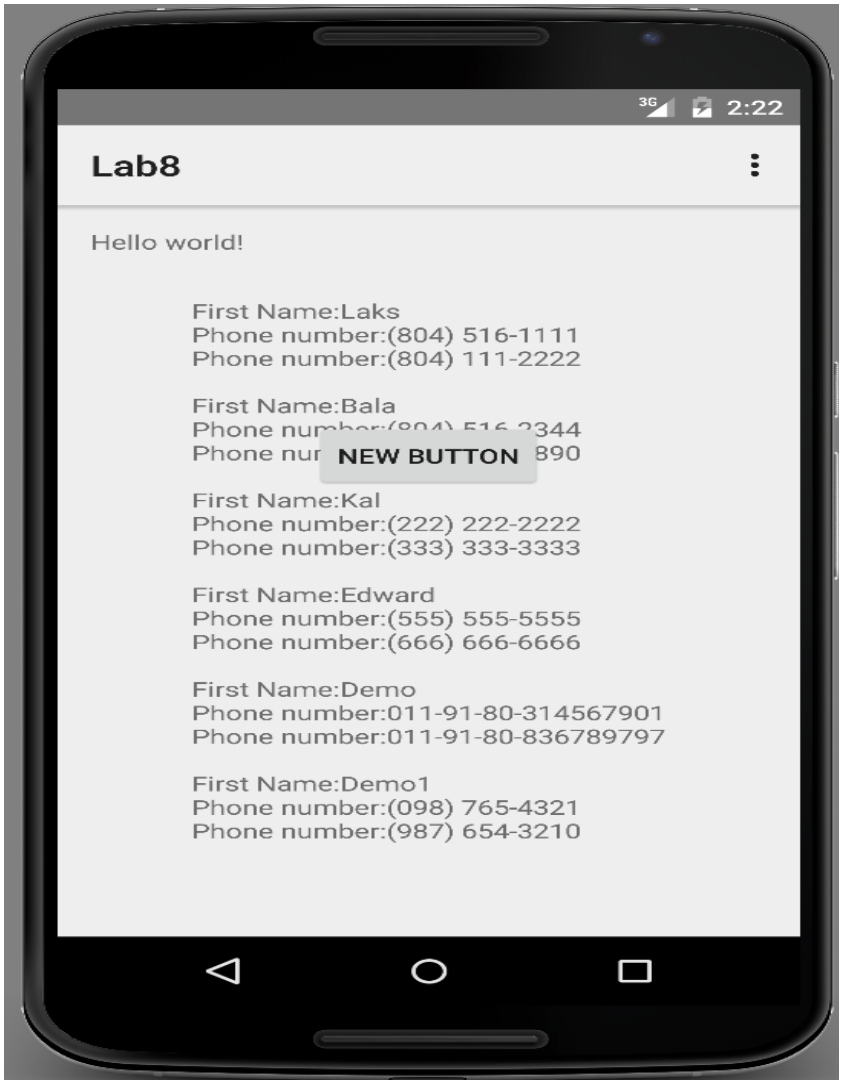
→



© 2015 IBM Corporation

Run the Application

- Click on New Button → Display all Contacts



© 2015 IBM Corporation

Asynchronous Task (AsyncTask) in Android

© 2015 IBM Corporation

AsyncTask

- One of the most important problems in Android development is the User Interface (UI) responsiveness. Part of this is the fault of Android implementation, but much faults lies on the developer.

- A key to a good GUI performance in Android development is to move as much work as possible off from the UI thread into the background. Using AsyncTasks is one of the easiest way to do this.

When to use AsyncTask?

An AsyncTask is designed to be a helper class around threads and handlers. As described in the Android documentation:

> *"AsyncTask enables proper and easy use of the UI thread. This class allows to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers."*

Simply said: An AsyncTask is used to do some computation in background and publish the result on the UI thread.

© 2015 IBM Corporation

AsyncTask – Cont'd

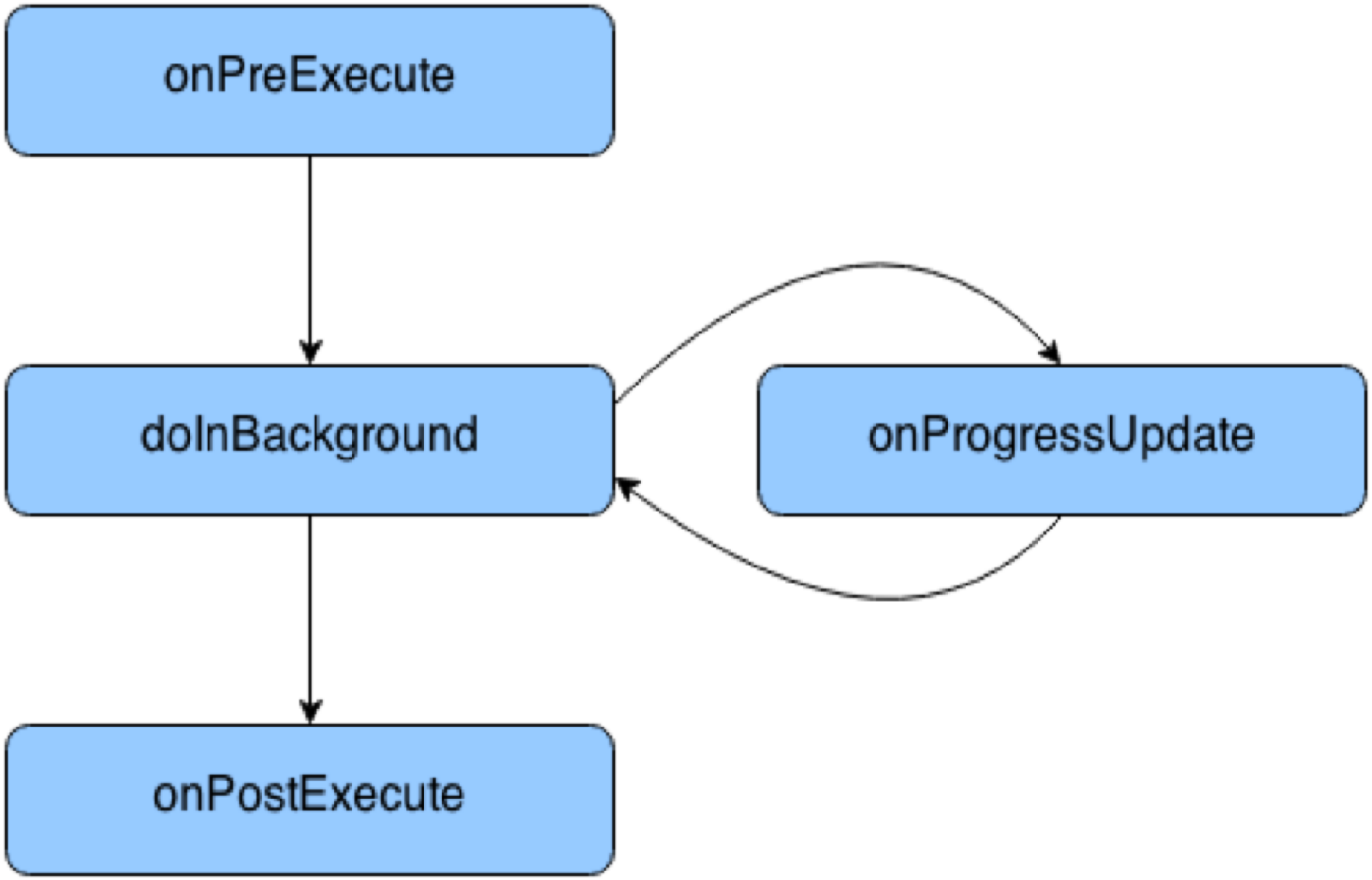
- AsyncTasks ideally are designed to perform operations in background which take at most a few seconds. **It is not recommended to use them for downloading megabytes of data from servers or doing CPU intensive tasks, such as image processing.**

- If you want to keep threads running for a long period of time, it is strongly recommended to use Java threads or classes such as java.util.concurrent.Executor or java.util.concurrent.ThreadPoolExecutor in combination with android.os.Handler to publish the results on the UI thread.

Note : If you need to do network operations using either Java threads or AsyncTask. In Android 4 and above attempting to access the network from the UI thread will throw an exception.

© 2015 IBM Corporation

AsyncTask – Life Cycle



© 2015 IBM Corporation

AsyncTask – Life Cycle - Cont'd

Step	Thread	Description
onPreExecute	UI	This is the first step of execution of an AsyncTask and invoked on the UI thread. Normally it will be used to set up the task e.g. to instantiate variables.
doInBackground	Background	Immediately after onPreExecute this method will be invoked on the background thread. The background computation that can take a long time should be performed in this method. The computation's result will be passed back to the last step. It is possible to call the publishProgress() method from this step, e.g. to make a update of the current state of the UI, which will be done by the onProgressUpdate() method.
onProgressUpdate	UI	This method is used to display any form of progress on the UI thread during the execution of background computation. For instance, it can be used to animate a progress bar or write logs on the text file.
onPostExecute	UI	This is the last step of the execution and invoked on the UI thread after the background computation has finished. The result of the background computation is passed to this step as a parameter.

© 2015 IBM Corporation

AsyncTask – Usage

- To use AsyncTask you must subclass it. An AsyncTask is started via the execute() method.
- The execute() method calls the doInBackground() method.
- AsyncTask is an intelligent thread that is recommended to be used. It is intelligent as it can help with it's methods, and there are two methods that run on the UI thread, which is good to update UI components.

Use AsyncTask for:

- Simple network operations which do not require downloading a lot of data
- Disk-bound tasks that might take less than a few milliseconds

© 2015 IBM Corporation

Lab9 - Click Counter Application Using AsyncTask

© 2015 IBM Corporation

Click Counter Application – User Interface

app
Lab9 > app > src > main > res > layout > activity_main.xml

Android
app
manifests
java
ecodcnc.com.lab9
MainActivity
ecodcnc.com.lab9 (androidTest)
res
drawable
layout
activity_main.xml
menu
mipmap
values
Gradle Scripts

MainActivity.java
activity_main.xml

Palette
Layouts
FrameLayout
LinearLayout (Horizonta
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
ProgressBar (Horizonta
SeekBar
RatingBar
Spinner
WebView
Text Fields
Plain Text
Person Name
Password
Password (Numeric)

Nexus 4 | AppTheme | MainActivity | 22

5:10
Lab9
Hello world!
1000
NEW BUTTON

Component Tree
Device Screen
RelativeLayout
textView – @string/hello_world
textview2 – "info"
button – "New Button"
editText – "1000"

Properties

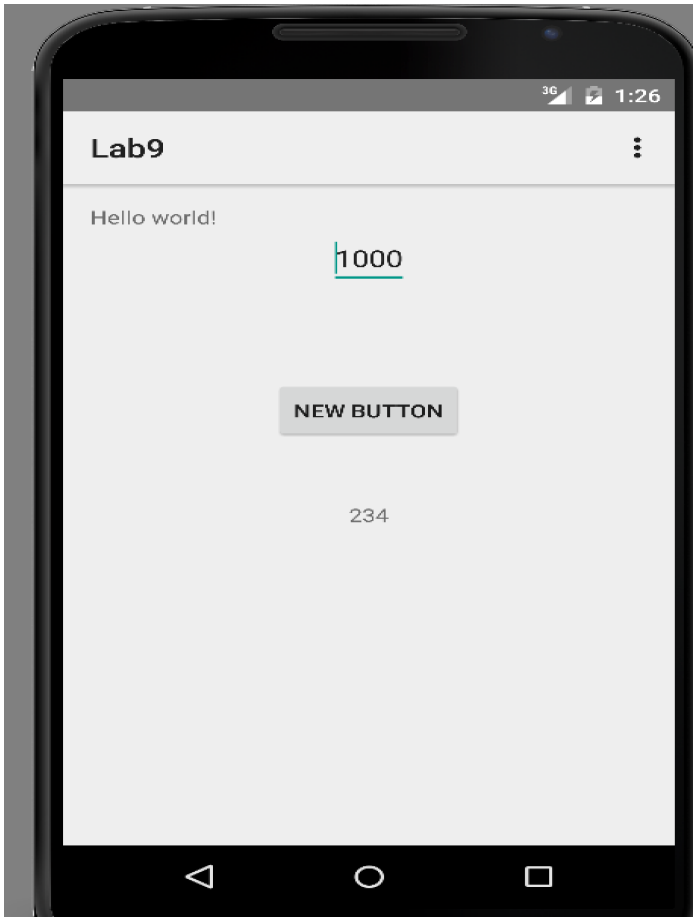
layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveRegion	
accessibilityTraversalA	
accessibilityTraversalB	
alpha	
background	
backgroundTint	
backgroundTintMode	
clickable	☐
contentDescription	
elevation	
focusable	☐
focusableInTouchMode	☐
gravity	[]

© 2015 IBM Corporation

Click Counter Application

```java
public class MainActivity extends Activity {

    EditText editText;
    TextView textView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void MyClick(View view) {

        Log.d("MyClick", "Yes working");
        BackGroundTask task = new BackGroundTask();
        task.execute(new Integer[]{10, 100, 1000});
    }

    private class BackGroundTask extends AsyncTask<Integer, Integer, String> {
        Integer progress = new Integer(0);

        @Override
        protected String doInBackground(Integer... args) {

            int result =0;
            for (int i=0;i<args[0];i++) { result=+i;}
            for (int i=0;i<args[1];i++) {result=+i;}
            for (int i=0;i<args[2];i++) {result=+i;}

            //simulate a long task by delaying for 5 seconds:
            long startTime = System.currentTimeMillis();
            long delay = 50000;
            long elapsedTime;
            while(System.currentTimeMillis() < startTime + delay) {
                elapsedTime = System.currentTimeMillis() - startTime;
                if( elapsedTime % 50 == 0) {
                    //update the progress bar...
                    progress = (int) elapsedTime / 50;
                    publishProgress(progress);
                }
            }
            //return a string representation of the square root:
            return String.valueOf(result);
        }
```

- **MainActivity → New Button Click → Executing a counter in the background invoking an AsyncTask**

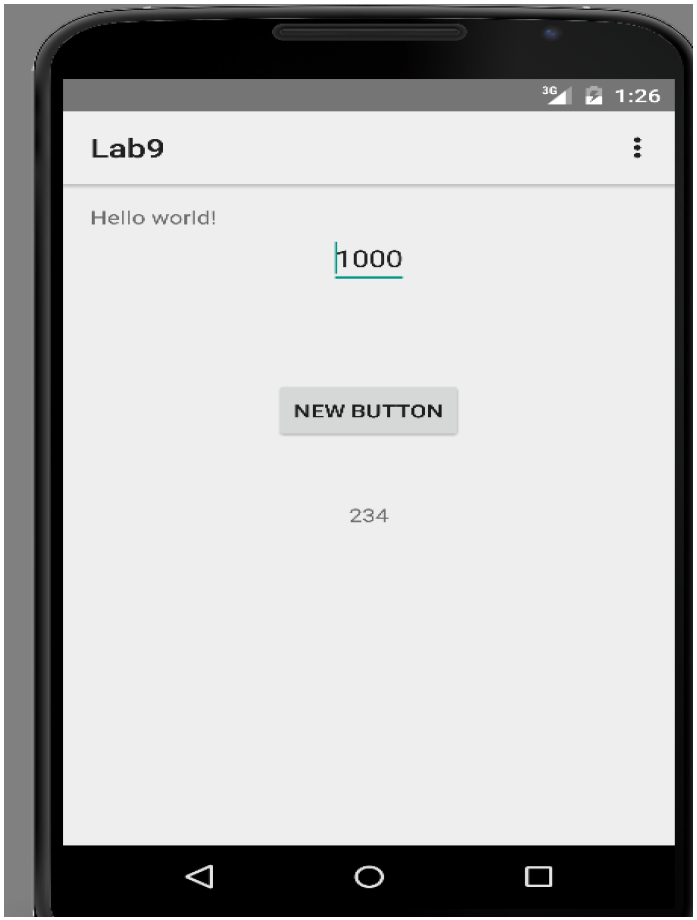


© 2015 IBM Corporation

Click Counter Application

```
private class BackGroundTask extends AsyncTask<Integer, Integer, String> {
    Integer progress = new Integer(0);

    @Override
    protected String doInBackground(Integer... args) {

        int result =0;
        for (int i=0;i<args[0];i++) { result=+i;}
        for (int i=0;i<args[1];i++) {result=+i;}
        for (int i=0;i<args[2];i++) {result=+i;}

        //simulate a long task by delaying for 5 seconds:
        long startTime = System.currentTimeMillis();
        long delay = 50000;
        long elapsedTime;
        while(System.currentTimeMillis() < startTime + delay) {
            elapsedTime = System.currentTimeMillis() - startTime;
            if( elapsedTime % 50 == 0) {
                //update the progress bar...
                progress = (int) elapsedTime / 50;
                publishProgress(progress);
            }
        }
        //return a string representation of the square root:
        return String.valueOf(result);
    }

    @Override
    protected void onProgressUpdate(Integer... units) {
        Log.i("HERE",Integer.toString(units[0]));
        textView = (TextView) findViewById(R.id.textview2);
        textView.setVisibility(View.VISIBLE);
        textView.setText(Integer.toString(units[0]));
    }

    @Override
    protected void onPostExecute(String result) {
        editText = (EditText) findViewById(R.id.editText);
        textView = (TextView) findViewById(R.id.textview2);
        textView.setVisibility(View.GONE);
        editText.setText(result);
    }
}
```

- **onProgressUpdate → Display the progress of the counter in AysncTask → on the screen below the button**



© 2015 IBM Corporation

Run the Click Counter Application

Lab9 > app > src > main > res > layout > activity_main.xml

Android

- app
  - manifests
  - java
    - ecodcnc.com.lab9
      - MainActivity
    - ecodcnc.com.lab9 (androidTest)
  - res
    - drawable
    - layout
      - activity_main.xml
    - menu
    - mipmap
    - values
  - Gradle Scripts

MainActivity.java | activity_main.xml

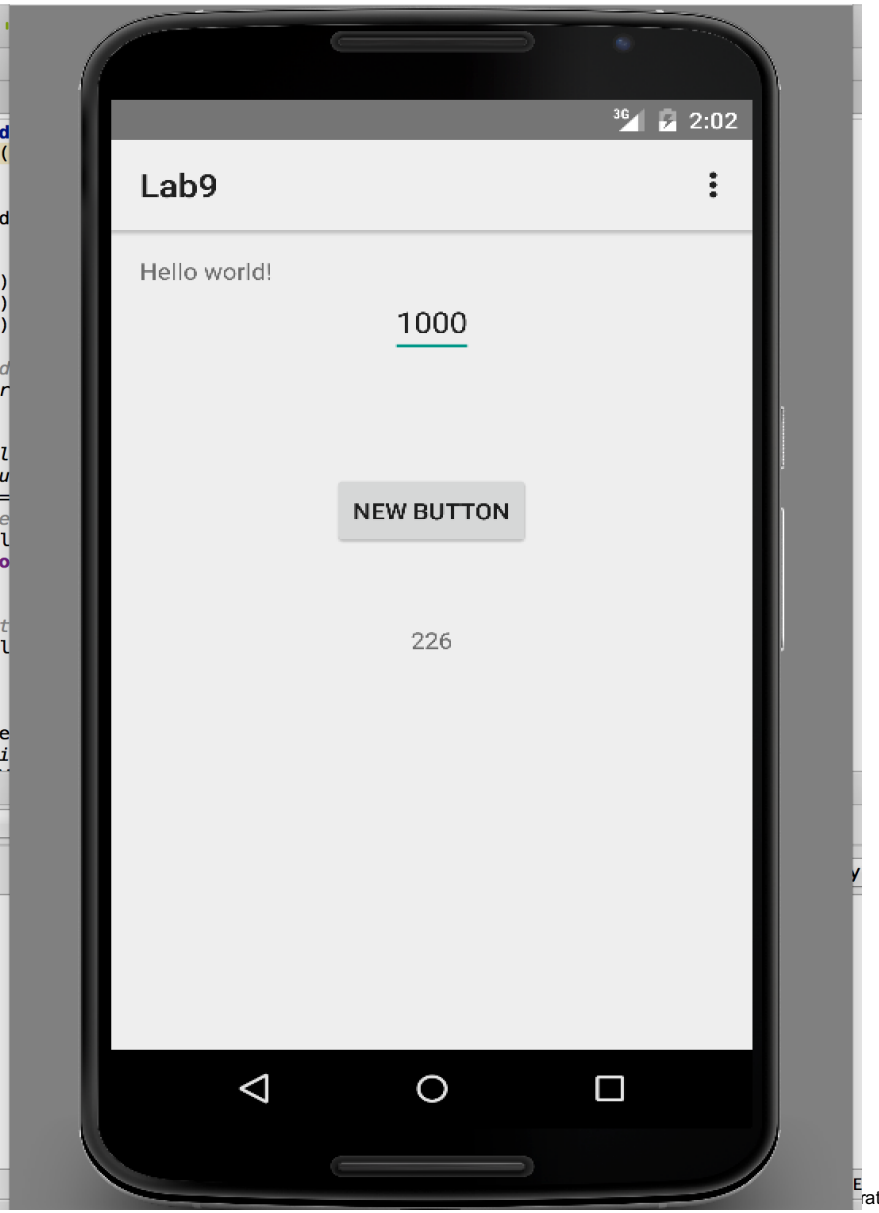
```
private class BackGroundTask extend
    Integer progress = new Integer(

    @Override
    protected String doInBackground

        int result =0;
        for (int i=0;i<args[0];i++)
        for (int i=0;i<args[1];i++)
        for (int i=0;i<args[2];i++)

        //simulate a long task by d
        long startTime = System.cur
        long delay = 50000;
        long elapsedTime;
        while(System.currentTimeMil
            elapsedTime = System.cu
            if( elapsedTime % 50 ==
                //update the progre
                progress = (int) el
                publishProgress(pro
            }
        }
        //return a string represent
        return String.valueOf(resul
    }

    @Override
    protected void onProgressUpdate
        Log.i("HERE",Integer.toStri
```

Emulator Nexus_6_API_22 Android 5.1 (API 22) | ecodcnc.com.lab9 (1826)

logcat | ADB logs | Memory | CPU

```
09-23 14:02:19.116   1826-1826/ecodcnc.com.lab9 I/HERE : 213
09-23 14:02:19.166   1826-1826/ecodcnc.com.lab9 I/HERE : 214
09-23 14:02:19.216   1826-1826/ecodcnc.com.lab9 I/HERE : 215
09-23 14:02:19.287   1826-1826/ecodcnc.com.lab9 I/HERE : 216
09-23 14:02:19.287   1826-1826/ecodcnc.com.lab9 I/HERE : 216
09-23 14:02:19.287   1826-1826/ecodcnc.com.lab9 I/HERE : 216
09-23 14:02:19.317   1826-1826/ecodcnc.com.lab9 I/HERE : 217
09-23 14:02:19.366   1826-1826/ecodcnc.com.lab9 I/HERE : 218
09-23 14:02:19.416   1826-1826/ecodcnc.com.lab9 I/HERE : 219
09-23 14:02:19.466   1826-1826/ecodcnc.com.lab9 I/HERE : 220
09-23 14:02:19.566   1826-1826/ecodcnc.com.lab9 I/HERE : 222
09-23 14:02:19.616   1826-1826/ecodcnc.com.lab9 I/HERE : 223
```

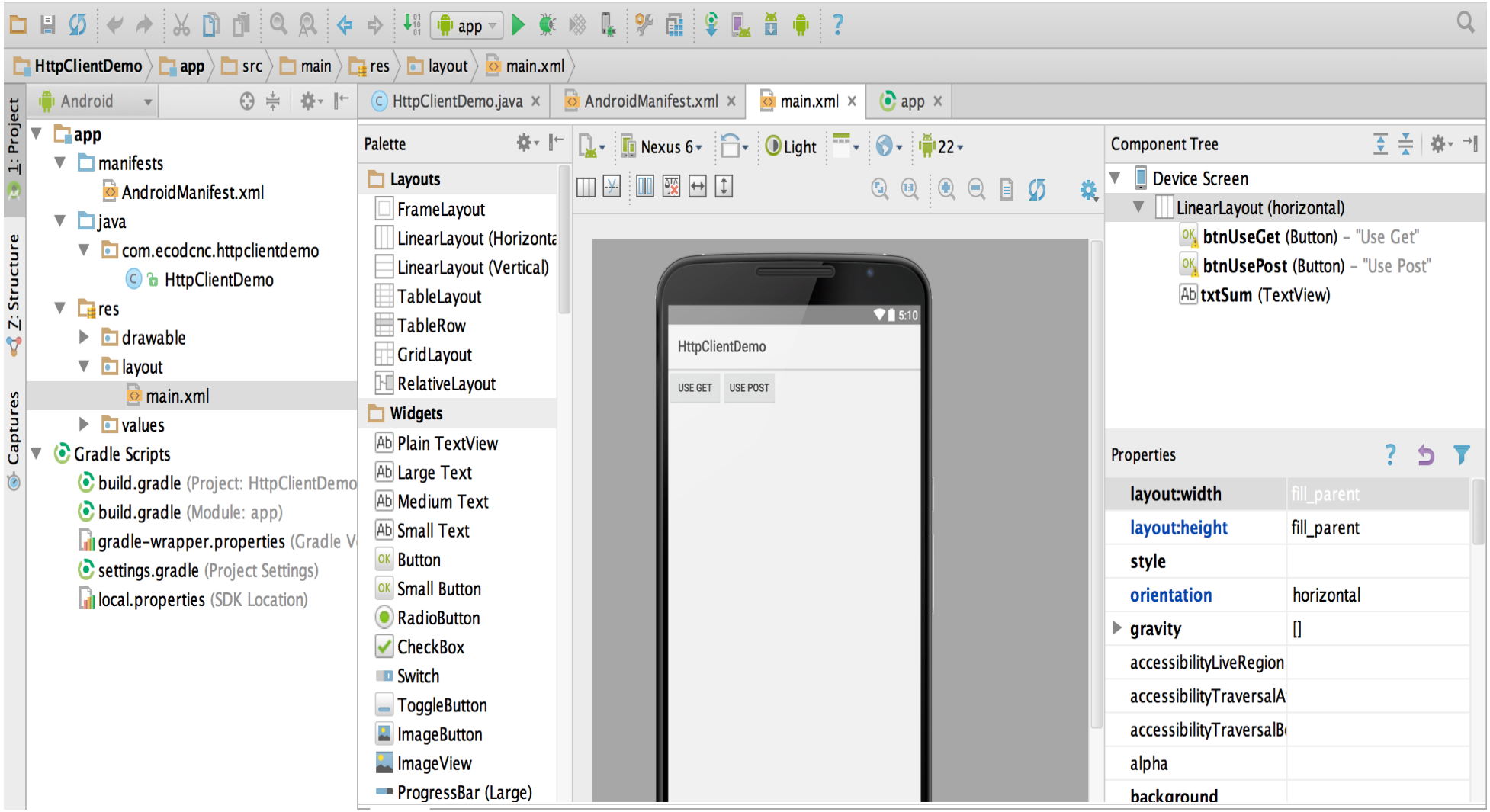
4: Run | TODO | 6: Android | Terminal | 0: Messages



Lab10 - HttpClientDemo

© 2015 IBM Corporation

Http Client Demo Application – User Interface

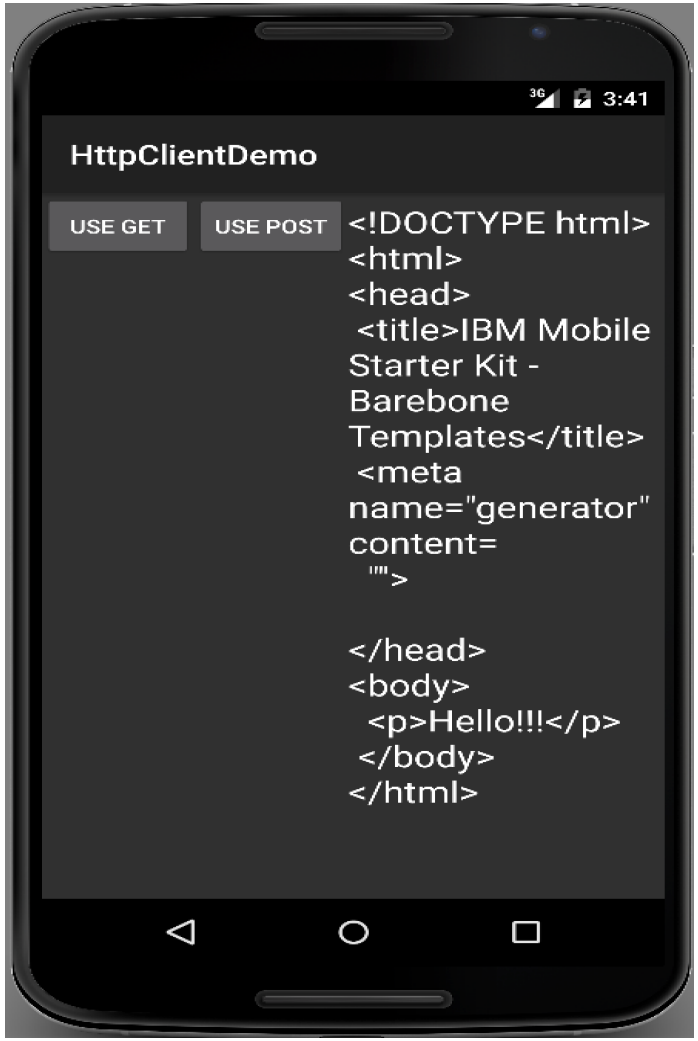


© 2015 IBM Corporation

HttpClientDemo Application - GET

```
public void useGet(View v) throws Exception {
    String response = new String();
    try {
        URL url = new URL(HOST);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        int HttpResult = conn.getResponseCode();

        // read the response
        System.out.println("Response Code: " + conn.getResponseCode());
        if (HttpResult == HttpURLConnection.HTTP_OK) {
            response = readResponse(conn.getInputStream());
        }else {
            response = conn.getResponseMessage();
            System.out.println(conn.getResponseMessage());
        }
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (ProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        tvSum.setText(response);
    }
}
```

```
private static final String HOST = "http://javaplays-restjava-mobilestarterkit-template.mybluemix.net/";
```



HttpClientDemo Application - POST

```
public void usePost(View v) {
    String response = new String();
    try {
        URL object = new URL(HOST + "api/service/login");
        HttpURLConnection con = (HttpURLConnection) object.openConnection();
        con.setDoOutput(true);
        con.setDoInput(true);
        con.setRequestProperty("Content-Type", "application/json");
        con.setRequestProperty("Accept", "application/json");
        con.setRequestMethod("POST");

        JSONObject cred = new JSONObject();
        cred.put("user_id", "admin");
        cred.put("password", "password");
        OutputStreamWriter wr = new OutputStreamWriter(con.getOutputStream());

        wr.write(cred.toString());
        wr.flush();

        //display what returns the POST request
        int HttpResult = con.getResponseCode();
        if (HttpResult == HttpURLConnection.HTTP_OK) {
            response = readResponse(con.getInputStream());
            System.out.println("" + response.toString());
        } else {
            response = con.getResponseMessage();
            System.out.println(con.getResponseMessage());
        }
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (ProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        tvSum.setText(response);
    }
```

```
private static final String HOST = "http://javaplays-restjava-mobilestarterkit-template.mybluemix.net/";
```

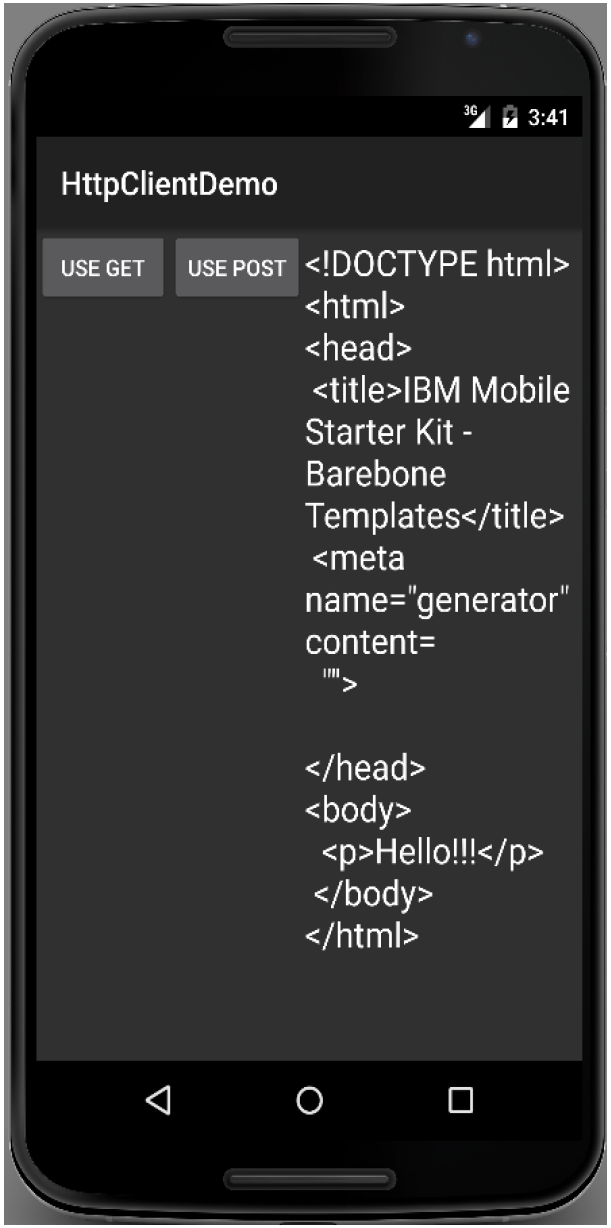
© 2015 IBM Corporation

HttpClientDemo Application – Parse RESPONSE (GET/POST)

```java
/**
 * processes an HttpResponse and returns a String
 *
 * @param res - an HttpResponse object
 * @return String representation of the response object
 */
private String readResponse(InputStream res) {

    StringBuffer response = new StringBuffer();
    BufferedReader input = null;


    try {
        BufferedReader br = new BufferedReader(new InputStreamReader(res, "utf-8"));

        String line = null;
        while ((line = br.readLine()) != null) {
            response.append(line + "\n");
        }
        br.close();
        System.out.println("" + response.toString());
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (input != null) {
            try {
                input.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return response.toString();
}
```

© 2015 IBM Corporation

Run HttpClientDemo Application

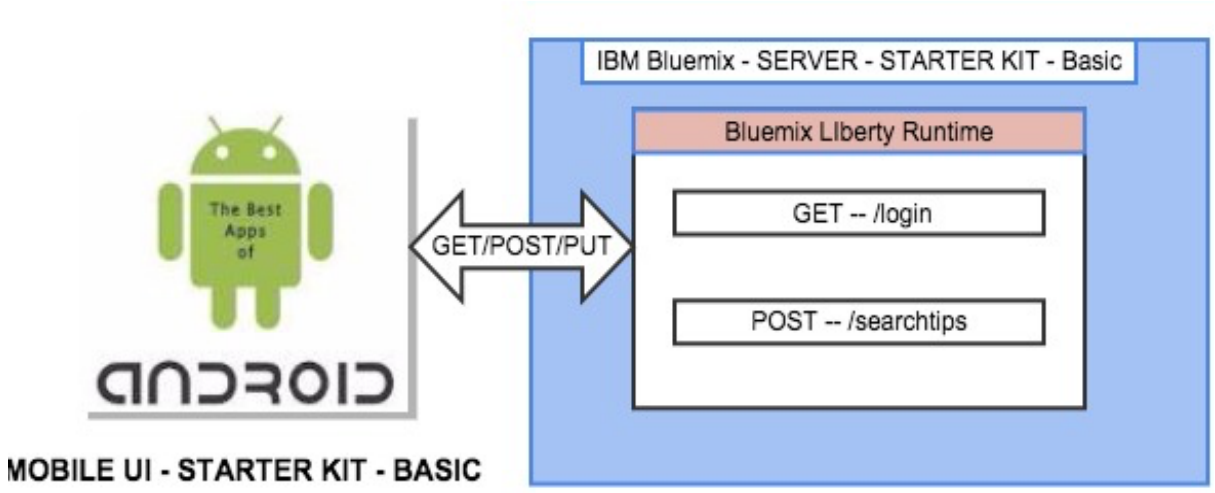


© 2015 IBM Corporation

Lab11 – MobileUIFrontEnd-BluemixBackend

© 2015 IBM Corporation

Android Mobile Application – Bluemix REST Service



Basic Application:

1) Create and Expose a Simple login REST Service, Searchtips Service in Bluemix
2) Create Simple UI Screens (Login, Search) and call REST Service in Bluemix

© 2015 IBM Corporation

Create Login UI Interface

javaplays-mobileui-mobilestarterkit-basic > res > layout > loginui.xml

Android

- javaplays-mobileui-mobilestarterk
  - manifests
  - java
  - res
    - drawable
    - layout
      - answer_list_item.xml
      - loginui.xml
      - outputui.xml
      - question_list_group.xml
      - response_ui.xml
      - searchui.xml
    - menu
    - values
  - assets
    - starterkit.properties
- Gradle Scripts

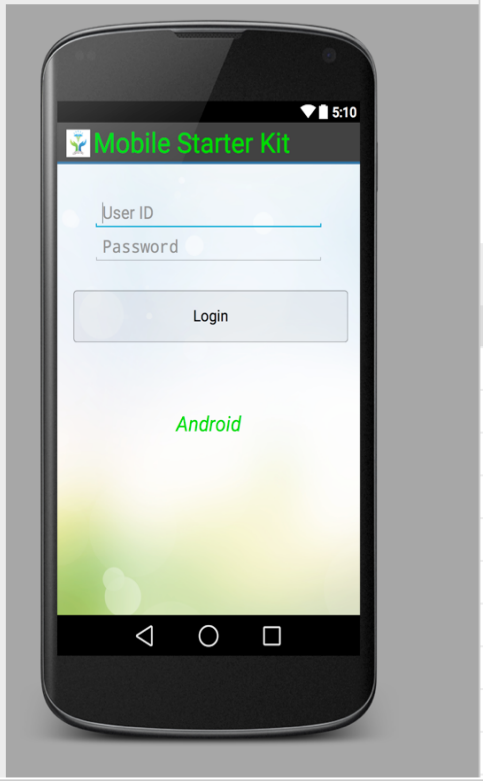
starterkit.properties | loginui.xml

Palette

Layouts
- FrameLayout
- LinearLayout (Horizonta
- LinearLayout (Vertical)
- TableLayout
- TableRow
- GridLayout
- RelativeLayout

Widgets
- Plain TextView
- Large Text
- Medium Text
- Small Text
- Button
- Small Button
- RadioButton
- CheckBox
- Switch
- ToggleButton
- ImageButton
- ImageView
- ProgressBar (Large)
- ProgressBar (Normal)
- ProgressBar (Small)
- ProgressBar (Horizonta

Nexus 4 | AppTheme | MainActivity | 22



Component Tree
- Device Screen
  - LinearLayout (vertical)
    - LinearLayout
    - RelativeLayout

Properties

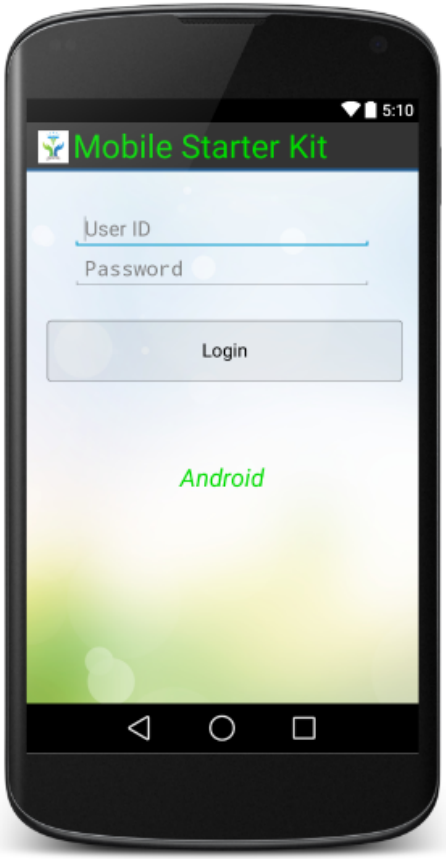
layout:width	match_parent
layout:height	match_parent
style	
orientation	vertical
gravity	[]
accessibilityLiveRegion	
accessibilityTraversalA	
accessibilityTraversalB	
alpha	
background	@drawable/bg
backgroundTint	

© 2015 IBM Corporation

Add Activity to Login UI Interface → Button Event

```
public class LoginUI extends Activity  implements TaskReceiver   {
    //key to read from the properties file; these have to match exactly as
    //in starterkit.properties file
    public static String _LOGIN_MSG_RESPONSE = "loginresponse";
    public static String _LOGIN_MSG_FAILED   = "loginfailedmessage";
    public static String _LOGIN_MSG_MISSING  = "loginparammissingmessage";
    public static final String CLASS_NAME = LoginUI.class.getSimpleName();
    private static String  _loginResponse            = null;
    private static String  _loginFailedMessage       = null;
    private  static String _loginParamMissingMessage = null;
    private ProgressDialog _pDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //read some values from the properties file
        readProperties();
        //TODO: to set the UI content here, uncomment the following line
        setContentView( R.layout.loginui);
        setBtnEvent();
    }
    @Override
    protected void onResume(){
        super.onResume();
        enterApplication();

    }
    private void setBtnEvent(){
        //TODO: Set the button's click event here
        //retrieve the login button from the UI by using the button's ID
        //retrieve the login button from the UI by using the button's ID
        Button bt_Login    = (Button)   findViewById( R.id.bt_Login    );
        _pDialog = new ProgressDialog( this );
        //set the button's click listener
        bt_Login.setOnClickListener((view) → {
        //execute the checkLogin()
        checkLogin();
        });
```

- **Login UI → New Button Click → Define Button Event to trigger CheckLogin()**

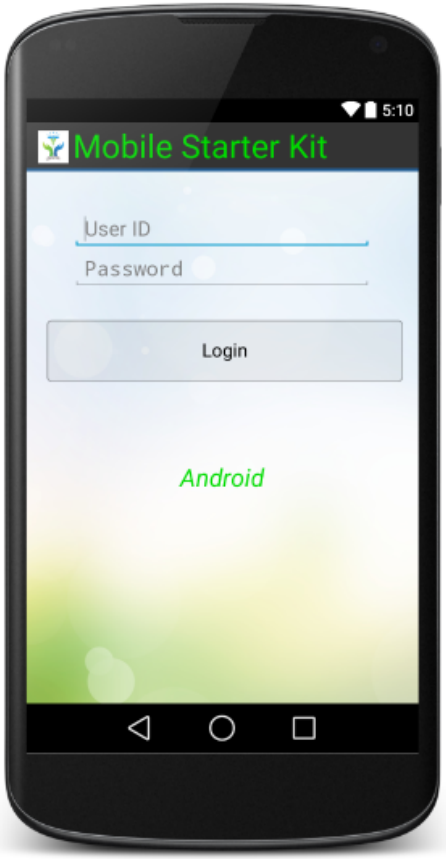


© 2015 IBM Corporation

Add Activity to Login UI Interface → CheckLogin()

```
/*
    Makes an Asynchronous call to the LoginTask to check the user's credential
 */
public void checkLogin(){

    //execute the login task asynchronously
    //retrieve the user's id and password
    //retreive the user's id and password
    EditText et_User_ID  = ( EditText ) findViewById ( R.id.et_UserID    );
    EditText et_Password = ( EditText ) findViewById ( R.id.et_Password  );

    String userID    = String.valueOf( et_User_ID.getText()  );
    String password  = String.valueOf( et_Password.getText() );

    if( userID.equals("") || password.equals( "" ) ){
        Toast.makeText( this, _loginParamMissingMessage, Toast.LENGTH_LONG ).show();
        return;
    }

    _pDialog.setTitle("Logging in..");
    _pDialog.setMessage("Please wait..");
    _pDialog.show();

    //Steps to make the call to an Aysnchronous task
    // 1. Create the task (the task class inherits the AsyncServiceTask class)
    LoginTask loginTask = new LoginTask( BlueMobileApp.getLoginURI( getApplicationContext() ) );

    // 2. Sets itself as the task's receiver
    loginTask.setReceiver(this);

    //3. Supply necessary parameters (String) and execute the task
    loginTask.execute(userID, password);
    return;

}
```

- **Login UI → CheckLogin() → Triggers LoginTask passing the URL**



© 2015 IBM Corporation

Define the REST URI and Messages for the Mobile App

starterkit.properties × | LoginUI.java × | LoginTask.java × | loginui.xml × | BlueMobileApp.java ×

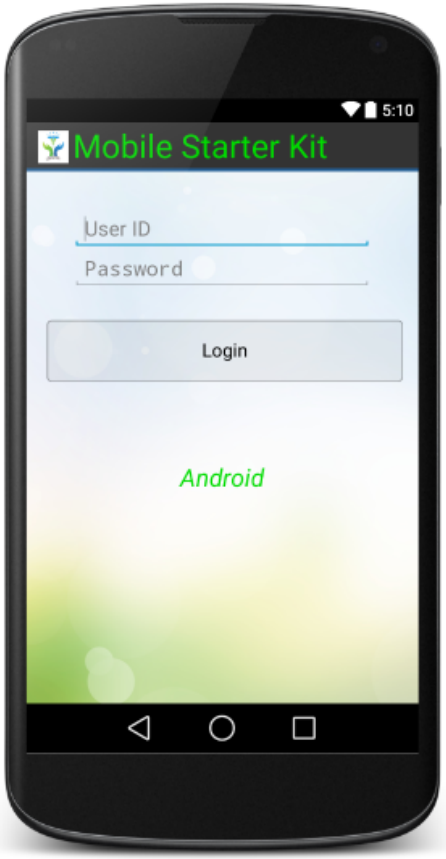
```
baseuri=http://javaplays-restjava-mobilestarterkit-template.mybluemix.net/api/service
loginuri=/login
searchuri=/searchtips
loginresponse=LoginResponse
loginfailedmessage="The login failed. Check the user id/password and try again
loginparammissingmessage=Please supply both the user id and password
searchloading=Loading..
searchmessage=Expanding concept and Collecting Tips..
searchnoresults=Please try again
```

© 2015 IBM Corporation

LoginTask (AsyncTask) → Performs Post to REST URI

```java
public class LoginTask  extends AsyncServiceTask {

    public static String _LOGIN_TASK_ID = "Log in";
    private String _loginURL ;

    public LoginTask( String loginURL ){
    //call the base class' constructor to assign a String
        //ID for this task
        super( _LOGIN_TASK_ID );
        _loginURL = loginURL ;
    }

    //override the base class' function to define the task
    @Override
    protected String performTask(String... params) throws IOException,
            JSONException, URISyntaxException {
        // call the server's login REST API
        //retrieve the user ID and password from the params...
        //retrieve the user ID and password from the params...
        String userID    = params[0];
        String password  = params[1];

        Log.i("LoginTask", userID);
        Log.i("LoginTask", password);
        //prepare a JSON object
        JSONObject credential = new JSONObject();
        credential.put( "user_id", userID );
        credential.put("password", password);
        //perform the post operation using the base class' utlity function
        return performPost( _loginURL , credential.toString(), null, null );
    }

}
```

- **LoginTask → Invokes performpost (HttpPost) → URL**

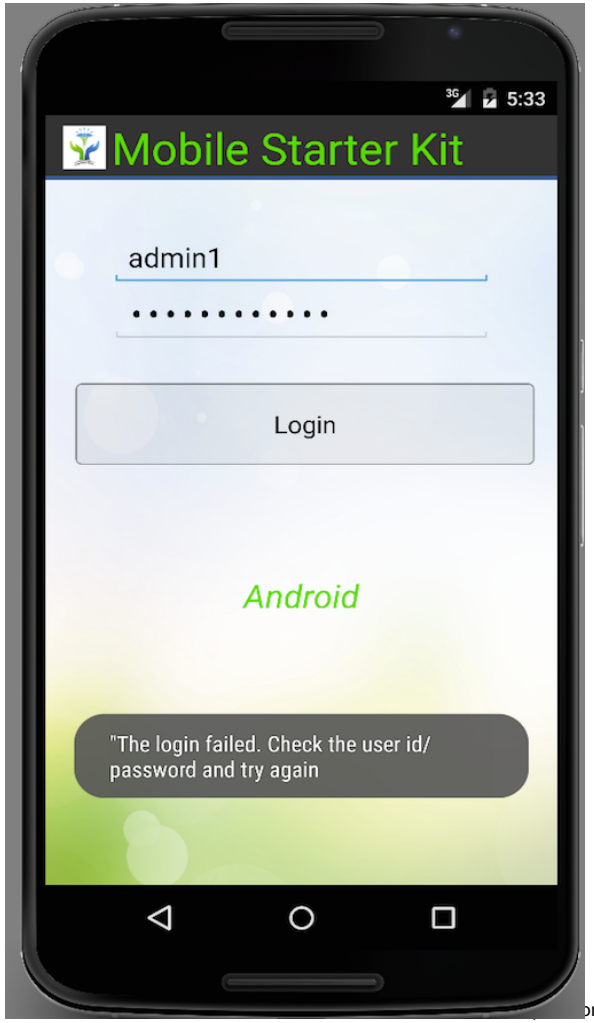


© 2015 IBM Corporation

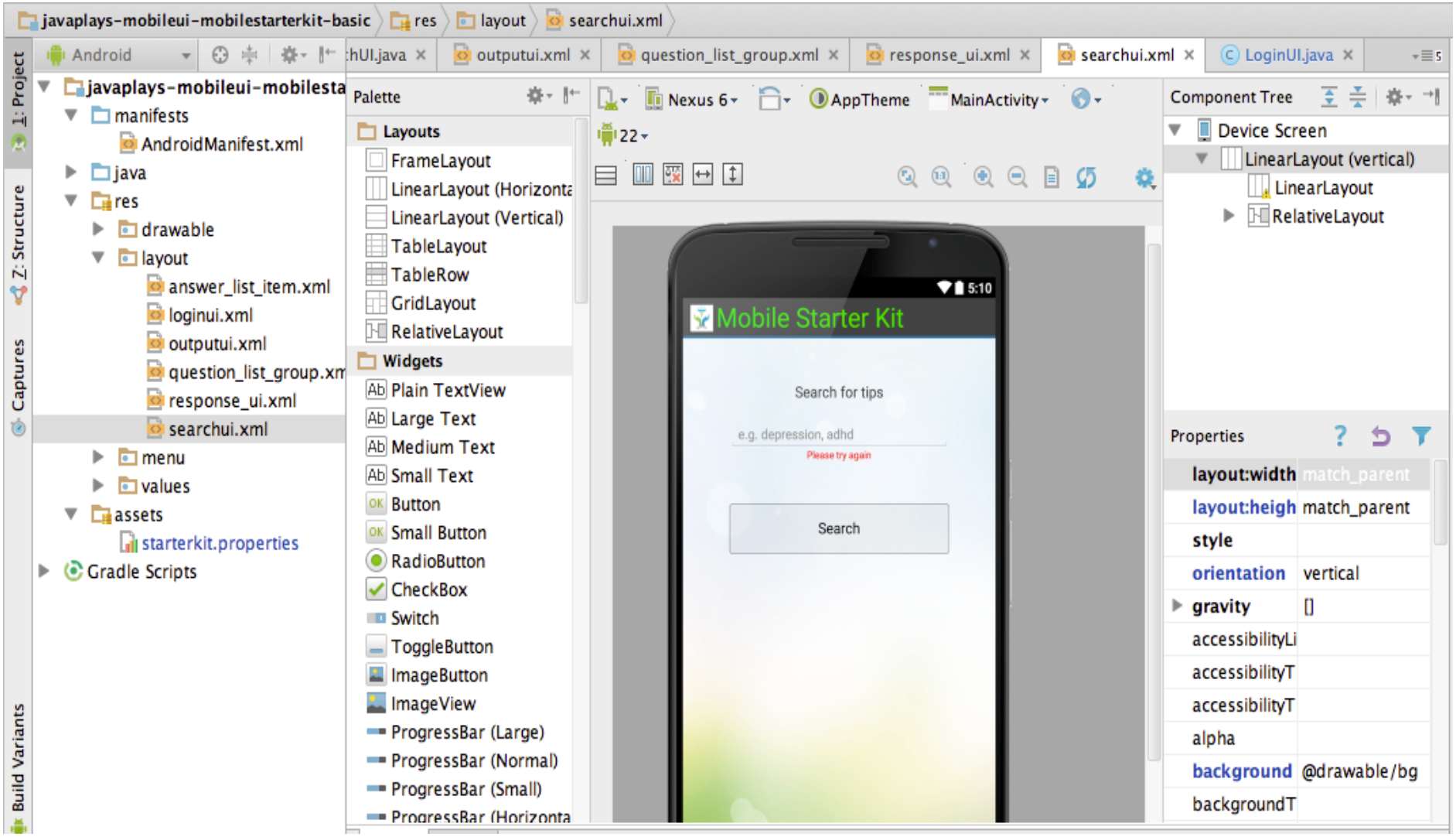
Login UI → ReceiveResult from LoginTask

```java
/*
    Implemented function of the TaskReceiver interface
    This is executed once the LoginTask is completed.
 */
@Override
public void receiveResult(String response, String source) {
    //TODO: do something once the credential is checked
    //and the response is available
    EditText et_UserID  = (EditText) findViewById( R.id.et_UserID   );
    Button bt_Login     = (Button)   findViewById( R.id.bt_Login    );

    Log.i(_loginResponse, response);

    _pDialog.dismiss();

    //first verify that the task completed is indeed the LoginTask
    if( source.equals(LoginTask._LOGIN_TASK_ID) ){

        bt_Login.setEnabled(true);
        et_UserID.setEnabled(true);

        //if the response is Successful, enter the application
        if( response.equals( "Successful" ) ){

            //retrieve and set the user id
            String userID = String.valueOf( et_UserID.getText() );
            BlueMobileApp.setUserID( userID );
            _pDialog.dismiss();
            enterApplication();
        }

        else{
            Toast.makeText( this, _loginFailedMessage, Toast.LENGTH_LONG).show();
            Log.i("Login Status" , "Failed " );
        }

    }

    return;
```

- **Login UI receives the results from LoginTask → Allows the user to next screen OR shows a validation error based on the credentials entered**



Create Search Interface

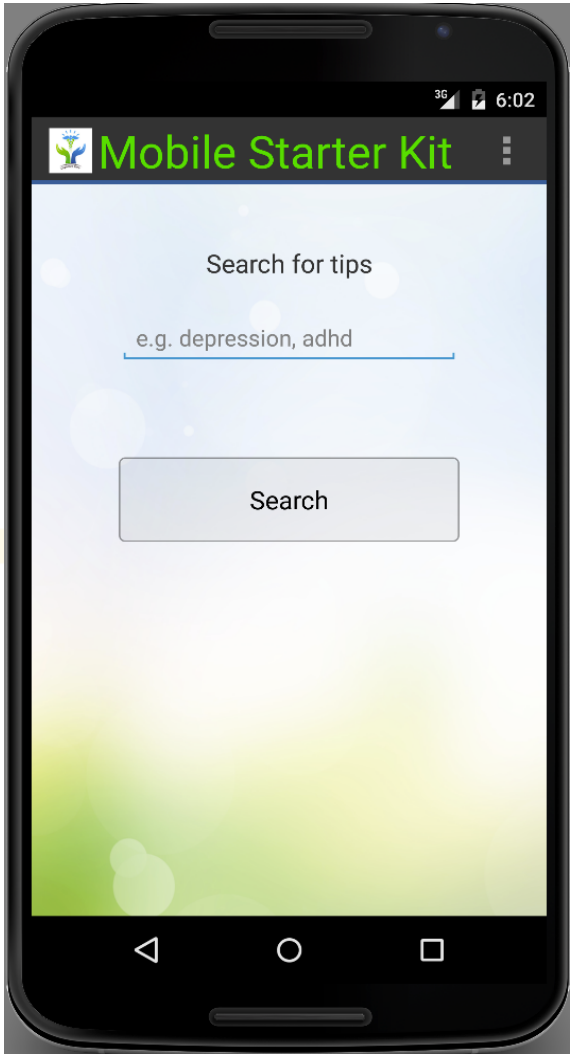


© 2015 IBM Corporation

Add Activity to Search UI Interface → Button Event

- Search → New Button Click → Define Button Event to trigger doSearch()

```
@Override
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate( savedInstanceState );

    //read some values from the properties file
    readProperties();

    _pDialog = new ProgressDialog(this);

    //sets the searchui.xml as its UI
    setContentView( R.layout.searchui );

    setupUIComponents();
}

private void setupUIComponents(){

    //make the error text box invisible and
    //setup the button's click event
    //retrieve the error textview and set it invisible
    TextView txtError = (TextView) findViewById( R.id.errorLabel );
    txtError.setVisibility(TextView.INVISIBLE);

    //retrieve the search button from the UI
    Button bt = ( Button ) findViewById( R.id.searchButton );

    //and set its click listener to execute the search Task
    bt.setOnClickListener( (v) → { doSearch(); });


    return;
}
```

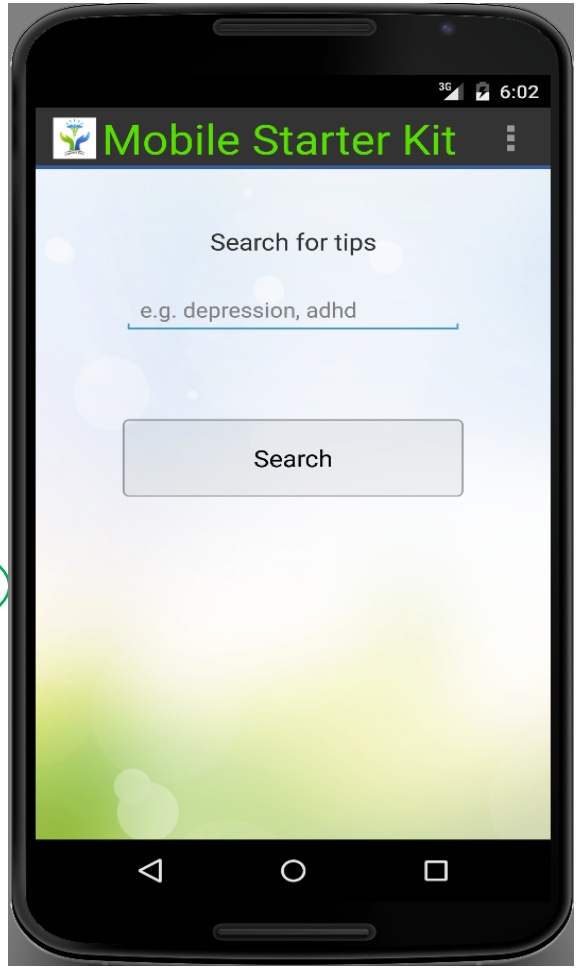


© 2015 IBM Corporation

Add Activity to Login UI Interface → CheckLogin()

```
/*
    Makes an Asynchronous call to the SearchTipsTask to find tips
 */
public void doSearch(){

    //execute the search task asynchronously
    TextView txtError = (TextView) findViewById( R.id.errorLabel );
    txtError.setVisibility(TextView.INVISIBLE);

    _pDialog.setCancelable(false);
    _pDialog.setTitle(_searchLoading);
    _pDialog.setMessage(_searchMessage);
    _pDialog.show();

    //retrieve the user input
    EditText et = ( EditText ) findViewById( R.id.inputKeyword );
    _seeds = et.getText().toString();

    //Steps to make the call to an Aysnchronous task
    // 1. Create the task (the task class inherits the AsyncServiceTask class)
    SearchTipsTask stt = new SearchTipsTask( BlueMobileApp.getSearchURI(getApplicationContext())
    // 2. Sets itself as the task's receiver
    stt.setReceiver( this );
    //3. Supply necessary parameters (String) and execute the task
    stt.execute(_seeds);

    Button   bt = ( Button )   findViewById( R.id.searchButton );
    bt.setEnabled(false);
    return;
}
```

- **Search UI→ doSearch() → Triggers SearchTipsTask passing the URL**

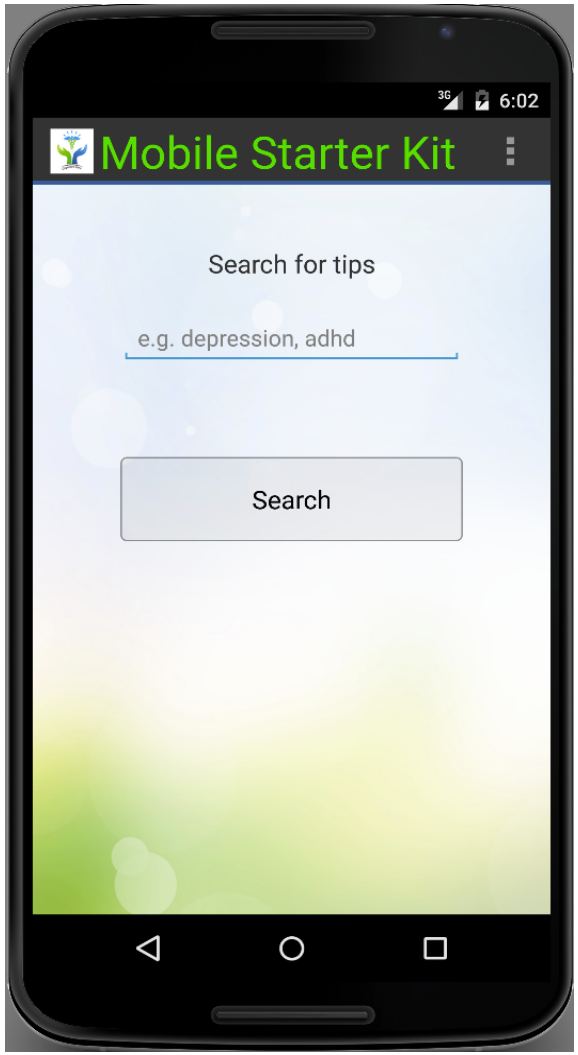


© 2015 IBM Corporation

SearchTipsTask (AsyncTask) → Performs Post to REST URI

```
public class SearchTipsTask  extends AsyncServiceTask {

    public static String _SEARCH_TIPS_ID = "Search Tips";

    private String _searchTipsURL;

    public SearchTipsTask( String searchURL ){

        //call the base class' constructor to assign a String
        //ID for this task
        super(_SEARCH_TIPS_ID);

        _searchTipsURL = searchURL;


    }

    @Override
    protected String performTask(String... params) throws IOException,
            JSONException, URISyntaxException {

        //TODO: call the server's searchtips REST API
        //retrieve the keyword from the params
        //retrieve the keyword from the params
        String keyword = params[0];

        //prepare a JSON object
        JSONObject searchParams = new JSONObject();
        searchParams.put("question", keyword);

        //perform the post operation using the base class' utility function
        return performPost( _searchTipsURL , searchParams.toString(), null, null );
    }
```

- **SerachTipsTask → Invokes performpost (HttpPost) → URL**

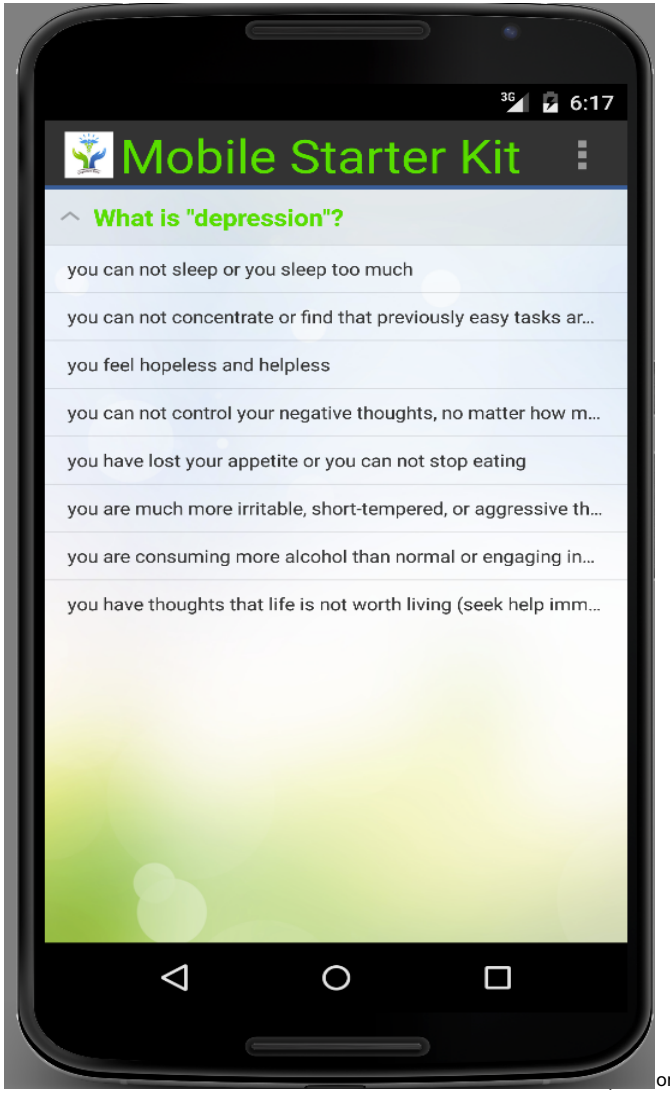


© 2015 IBM Corporation

Search UI → ReceiveResult from SearchTips

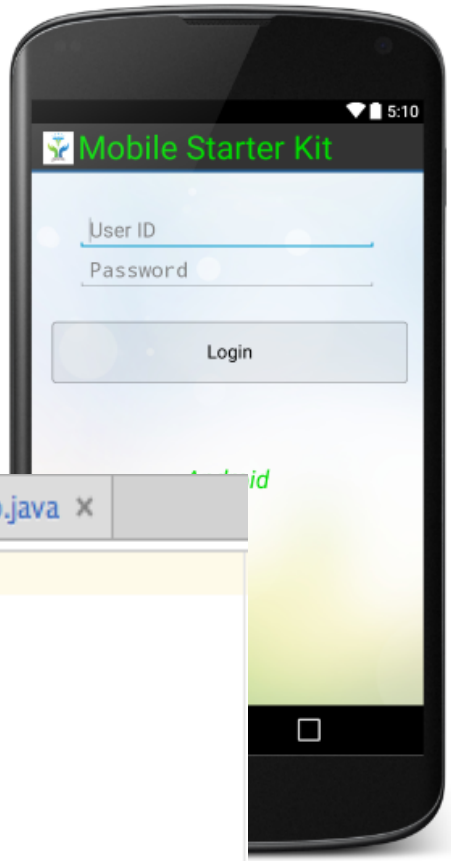
```
Implemented function of the TaskReceiver interface
This is executed once the SearchTipsTask is completed.
*/
@Override
public void receiveResult( String response, String source ) {
    //do something when the search response is available
    _pDialog.dismiss();
    TextView txtError = (TextView) findViewById( R.id.errorLabel );

    Button   bt = ( Button )   findViewById( R.id.searchButton );
    bt.setEnabled( true );

    //first verify that the task completed is indeed the SearchTipsTask
    if( source.equals( SearchTipsTask._SEARCH_TIPS_ID ) ){

        //if the task is not successful
        if( response.isEmpty() || response == null || response.equals( "Failed" ) ){
            txtError.setVisibility( TextView.VISIBLE );
            txtError.setText( _searchNoResults );
            return;
        }
        //if it is successful
        else{

            Log.i( "Response", response );
            txtError.setText( "" );
            txtError.setVisibility( TextView.INVISIBLE );

            //creat a new intent with the OutputUI
            Intent responseIntent = new Intent( SearchUI.this, OutputUI.class );
            //add some message for the OutputUI
            //first add what the tips are
            responseIntent.putExtra( BlueMobileApp._SEARCH_RESULT_MSG, response );
            //then add the original keyword
            responseIntent.putExtra(BlueMobileApp._KEYWORD_MSG, _seeds);

            //start the new activity
            startActivity(responseIntent);
        }
    }
    return;

}
```

- **Search UI receives the results from SearchTips → Searches all the results for the search**



Registering the interfaces

- At this stage, we have a working application ready to be built and run
  - Confirm the “baseuri” value is correct in assets->starterkit.properties file



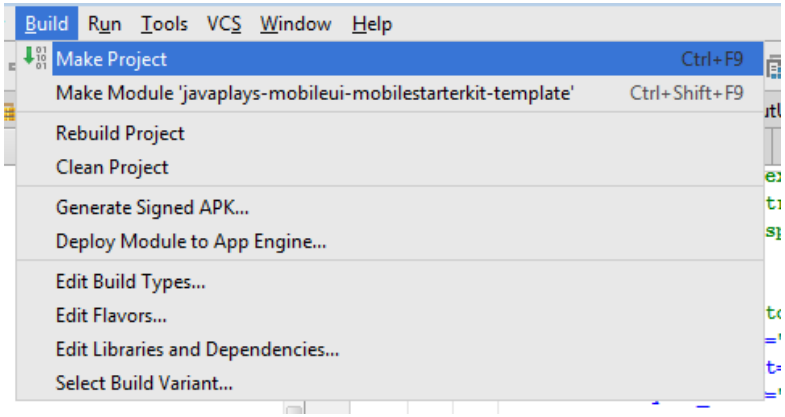
starterkit.properties | LoginUI.java | LoginTask.java | loginui.xml | BlueMobileApp.java

```
baseuri=http://javaplays-restjava-mobilestarterkit-template.mybluemix.net/api/service
loginuri=/login
searchuri=/searchtips
loginresponse=LoginResponse
loginfailedmessage="The login failed. Check the user id/password and try again
loginparammissingmessage=Please supply both the user id and password
searchloading=Loading..
searchmessage=Expanding concept and Collecting Tips..
searchnoresults=Please try again
```

© 2015 IBM Corporation

Build the Android Application

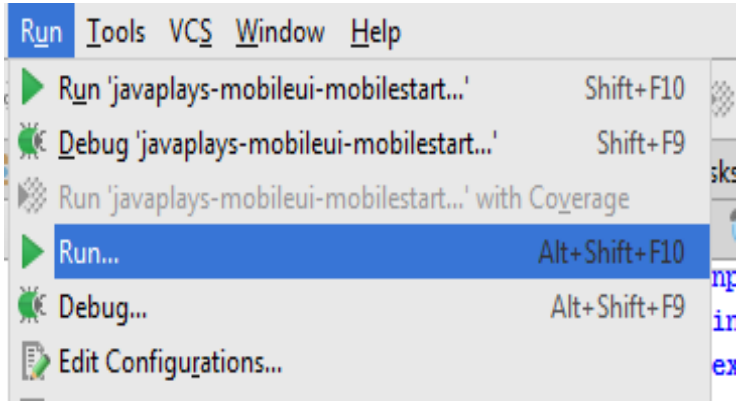
- Building and running Android Application

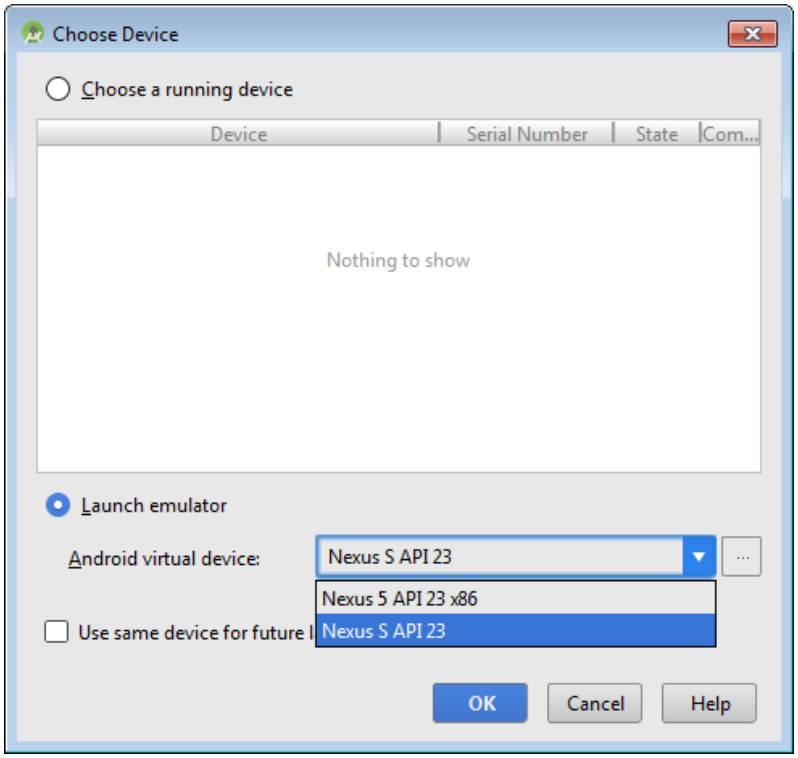


© 2015 IBM Corporation

Run the Android Application

- Running Android Application

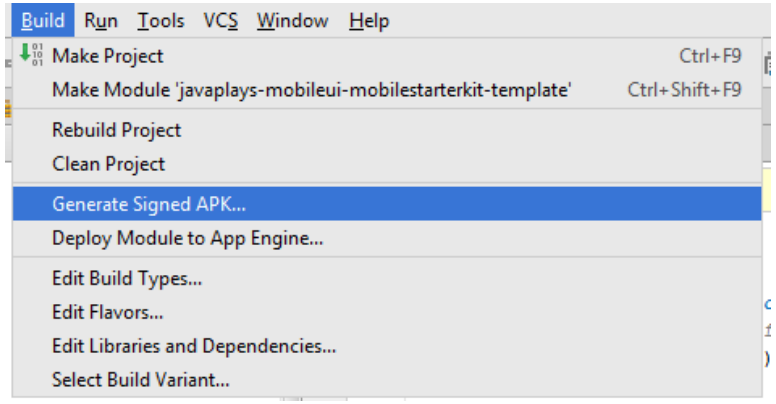


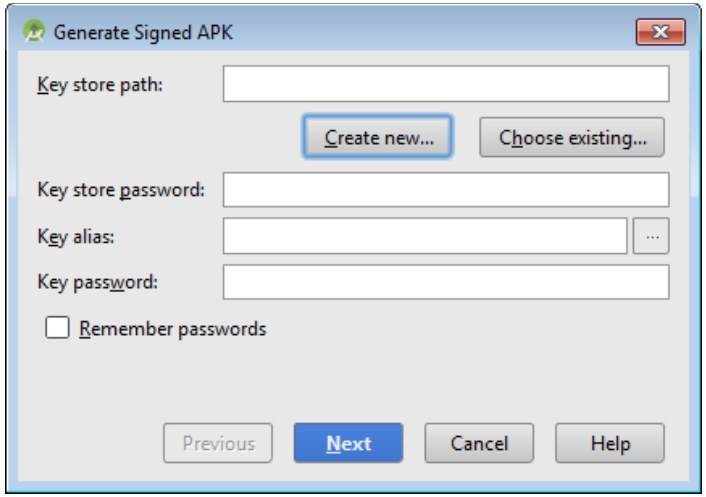


© 2015 IBM Corporation

Generating an APK Signing an Android Application

- From **Build** menu choose **"Generate Signed APK**
- From the dialogue box, click **Create New**
  - If you already have a keystore, please choose the **Choose Existing** and provide the necessary password

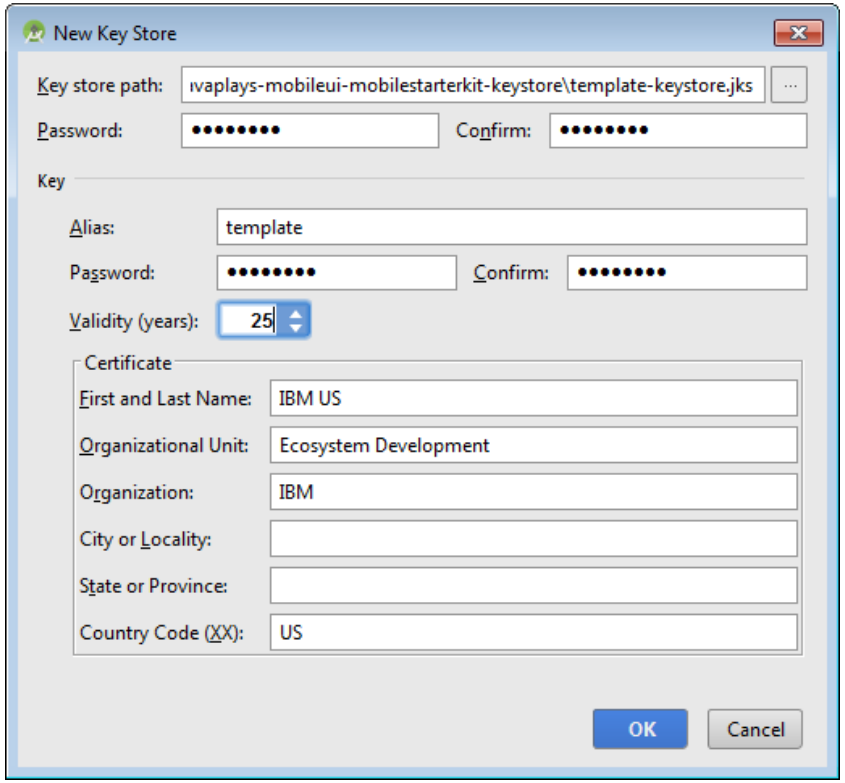




© 2015 IBM Corporation

Generating an APK Signing an Android Application

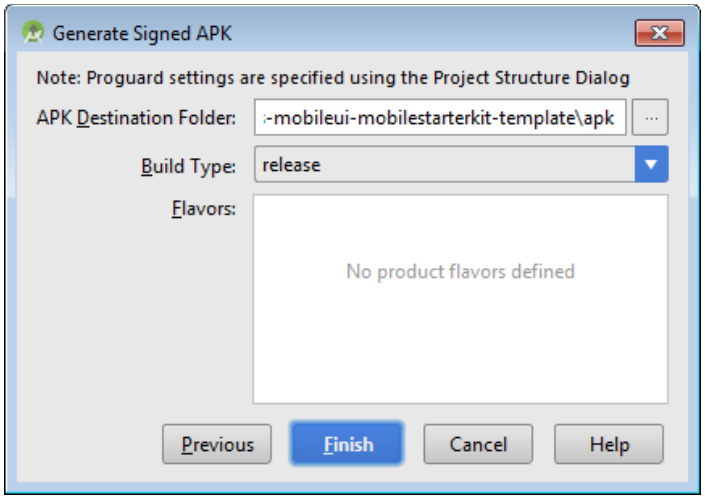
- Provide the Key Store Path
- Provide the key
  - In this example every password is “template”



© 2015 IBM Corporation

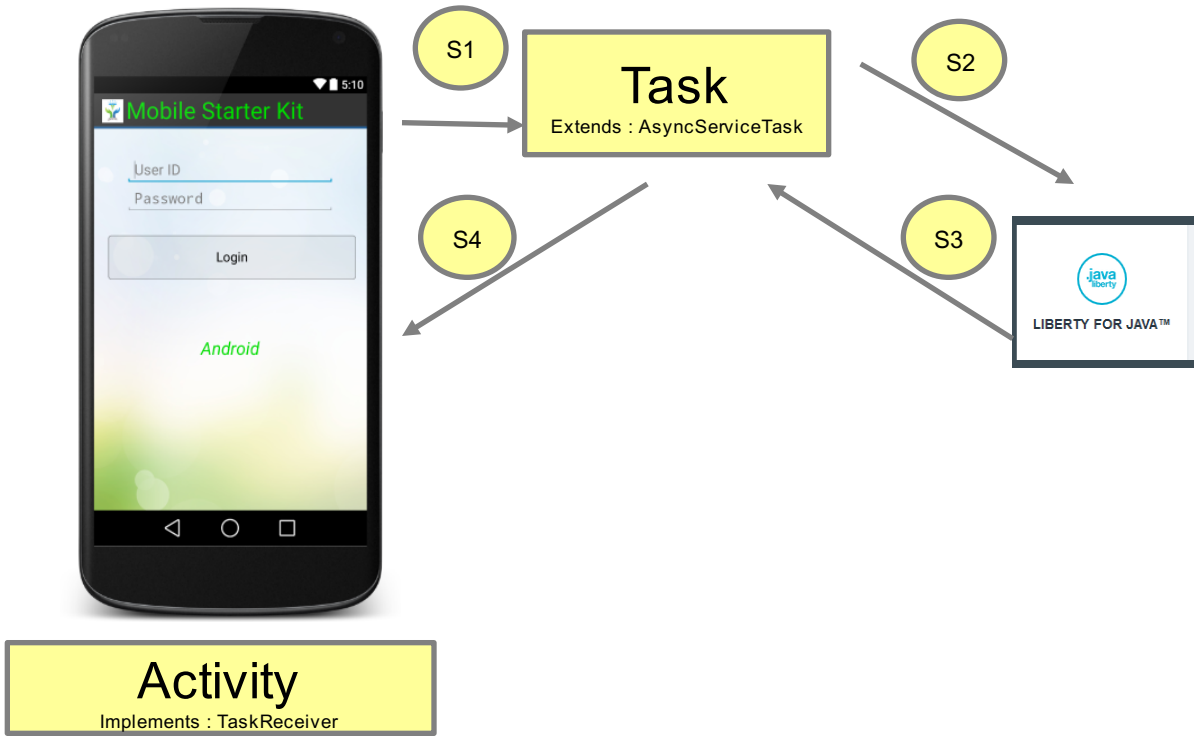
Signing an Android Application

- Once successful, the wizard will ask for a destination
- Click Finish
- The process will confirm the generation of the APK
  - Now your application is ready to be released



© 2015 IBM Corporation

Task Execution Protocol – Step by step



- S1 :
  - Activity creates the task and assigns an ID to it
  - Activity registers itself as a listener to the postExecution event of the task
  - Activity starts the task in a separate thread
- S2 :
  - Task executes a POST method and waits for the result
- S3 :
  - The response is received by the task thread
  - Task raises a postExecute() event and passes the response to any listener
- S4 :
  - Activity responds to the event and process the result

© 2015 IBM Corporation

Task Execution Protocol – Step by step

```
//SearchUI.java  - if the user clicks the button
{

...
SearchTipsTask  stt = new SearchTipsTask(  ... );
stt.setReceiver(this);
stt.execute(_seeds);
...

}
```

S1

```
//AsyncServiceTask.java
@Override
protected String doInBackground( String... params
) {

//performTask is implemented by the

//SearchTipsTask class
return performTask( params );
}
```

S2

S3

```
//AsyncServiceTask.java
@Override
protected void onPostExecute( String result ){
    _response = result;
    if( _tr != null ){
       //calls the receiver's receiveResult function
       //adds the serviceID here
       //in this case the receiver is SearchUI.java
_tr.receiveResult( _response,_serviceID);
}
    }
}
```

S4

```
@Override
public void receiveResult( String response,
                           String source ) {
{

 //make sure the task ID is the search task
 if( source.equals( correct_id ) ){
//perhaps do something with the response
 }
}
```

© 2015 IBM Corporation

How are all these stitched together? - without headache

- com.ibm.android.asynctask.TaskReceiver - Interface
  - Every activity that required to use a REST API implements this class
  - This class enforces the activities to declare a receiveResult function

```
public interface TaskReceiver{
public void receiveResult( String response, String source );
}
```

String response

- The response received from REST call

String Source

- The ID of the asynchronous task generating the response

```
//Example : SearchUI.java
public void receiveResult( String response, String source ) {

//make sure the task ID is the search task
if( source.equals( SearchTipsTask._SEARCH_DOC_ID ) ){

        //do something with the response
}
}
```

© 2015 IBM Corporation

How are all these stitched together? - without headache

- com.ibm.android.asynctask.AsyncServiceTask - Abstract
  - Every task that calls REST API extends this class
  - This class enforces the tasks to define a REST API call

```
public abstract String performTask( String... params)
        throws IOException, JSONException, URISyntaxException;
```

String... params

- Input parameter needed for the task

```
//Example : SearchTipsTask
@Override
protected String performTask(String... params) throws IOException,
                                JSONException, URISyntaxException {

//retrieve the parameter
String keyword = params[0];

//POST and return the response
return performPost( ... );
}
```

© 2015 IBM Corporation

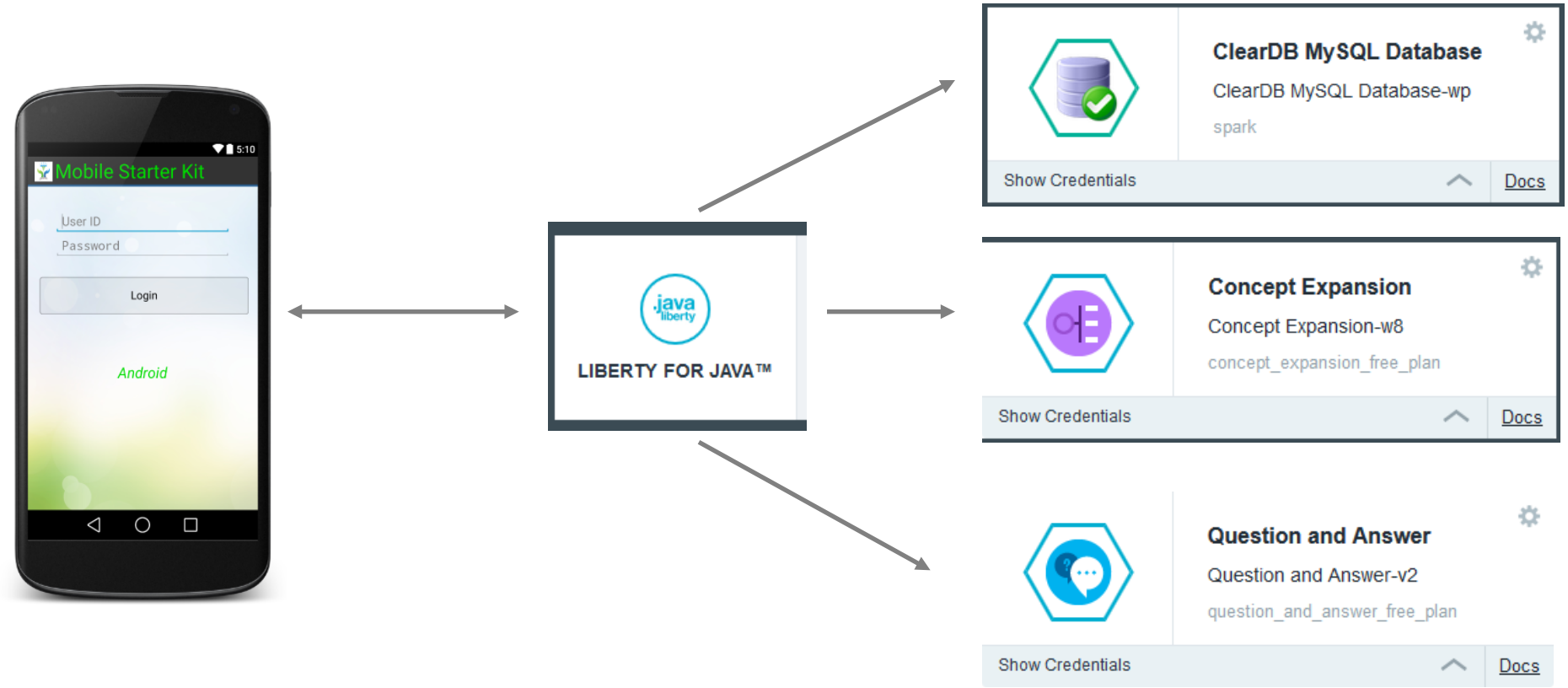
Lab12 – MobileUIFrontEnd-BluemixWatsonBackend

© 2015 IBM Corporation

Extend Mobile Application – Using Full Power of Bluemix by Adding Watson Services

© 2015 IBM Corporation

Light UI – Heavy Server side

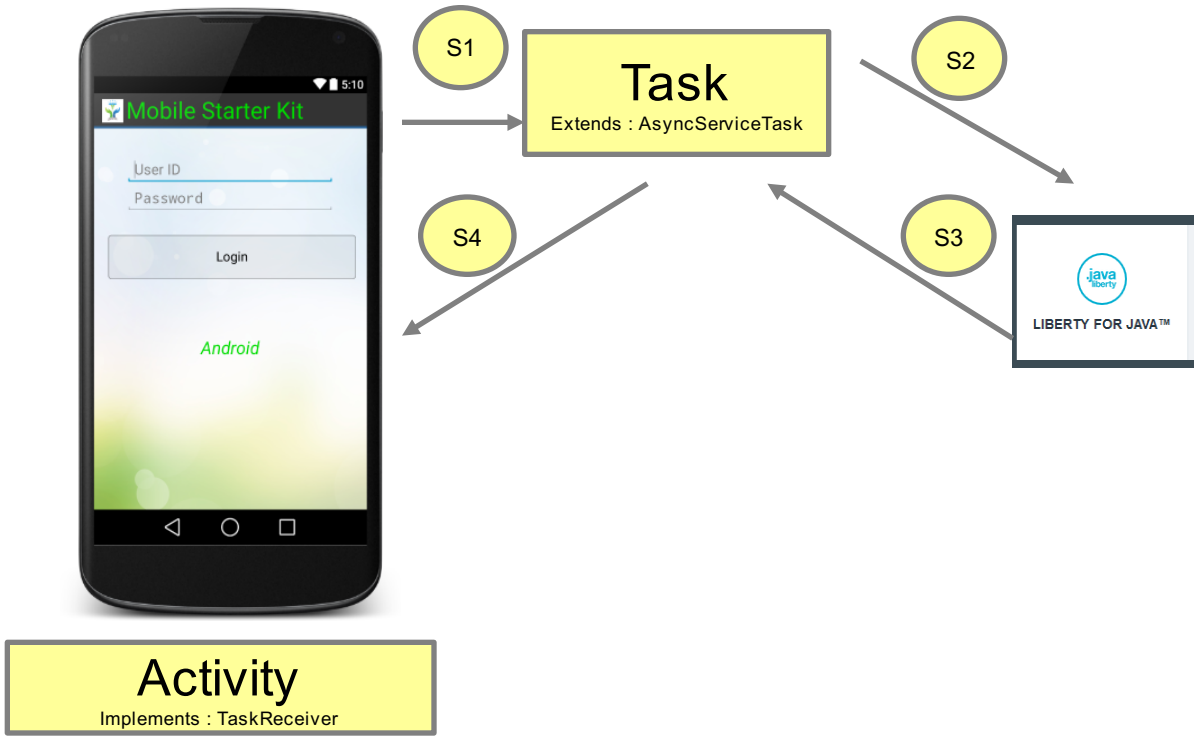


© 2015 IBM Corporation

Mobile UI Flow

© 2015 IBM Corporation

Task Execution Protocol – Step by step



- S1 :
  - Activity creates the task and assigns an ID to it
  - Activity registers itself as a listener to the postExecution event of the task
  - Activity starts the task in a separate thread
- S2 :
  - Task executes a POST method and waits for the result
- S3 :
  - The response is received by the task thread
  - Task raises a postExecute() event and passes the response to any listener
- S4 :
  - Activity responds to the event and process the result

© 2015 IBM Corporation

Task Execution Protocol – Step by step

```
//SearchUI.java  - if the user clicks the button
{

...
SearchTipsTask  stt = new SearchTipsTask(  ... );
stt.setReceiver(this);
stt.execute(_seeds);
...

}
```

S1

```
//AsyncServiceTask.java
@Override
protected String doInBackground( String... params ) {

//performTask is implemented by the
//SearchTipsTask class
return performTask( params );
}
```

S2

S3

```
//AsyncServiceTask.java
@Override
protected void onPostExecute( String result ){
    _response = result;
   if( _tr != null ){
      //calls the receiver's receiveResult function
      //adds the serviceID here
      //in this case the receiver is SearchUI.java
_tr.receiveResult( _response,_serviceID);
}
   }
}
```

S4

```
@Override
public void receiveResult( String response,
                           String source ) {
{

 //make sure the task ID is the search task
 if( source.equals( correct_id ) ){
//perhaps do something with the response
 }
}
```

© 2015 IBM Corporation

How are all these stitched together? - without headache

- com.ibm.android.asynctask.TaskReceiver - Interface
  - Every activity that required to use a REST API implements this class
  - This class enforces the activities to declare a receiveResult function

```
public interface TaskReceiver{
public void receiveResult( String response, String source );
}
```

String response

- The response received from REST call

String Source

- The ID of the asynchronous task generating the response

```
//Example : SearchUI.java
public void receiveResult( String response, String source ) {

//make sure the task ID is the search task
if( source.equals( SearchTipsTask._SEARCH_DOC_ID ) ){

        //do something with the response
}
}
```

© 2015 IBM Corporation

How are all these stitched together? - without headache

- com.ibm.android.asynctask.AsyncServiceTask - Abstract
  - Every task that calls REST API extends this class
  - This class enforces the tasks to define a REST API call

```
public abstract String performTask( String... params)
        throws IOException, JSONException, URISyntaxException;
```

String... params

- Input parameter needed for the task

```
//Example : SearchTipsTask
@Override
protected String performTask(String... params) throws IOException,
                                JSONException, URISyntaxException {

//retrieve the parameter
String keyword = params[0];

//POST and return the response
return performPost( ... );
}
```

© 2015 IBM Corporation