Android

Basics of UI Applications

Bibliography

1. Wallace Jackson, *Android Apps for Absolute Beginners*, Apress, 2017
2. Peter Späth, *Learn Kotlin for Android Development,* Apress 2019
3. Android Application Fundamentals, http://developer.android.com/guide/topics/fundamentals.html

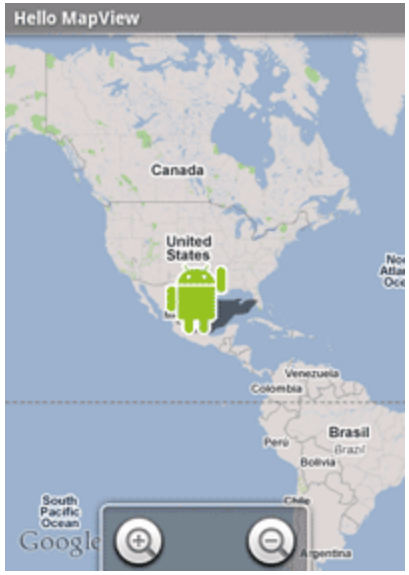
Contents

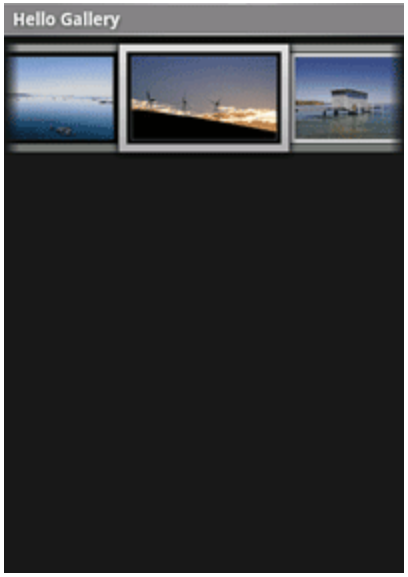
PDM
sayHello();

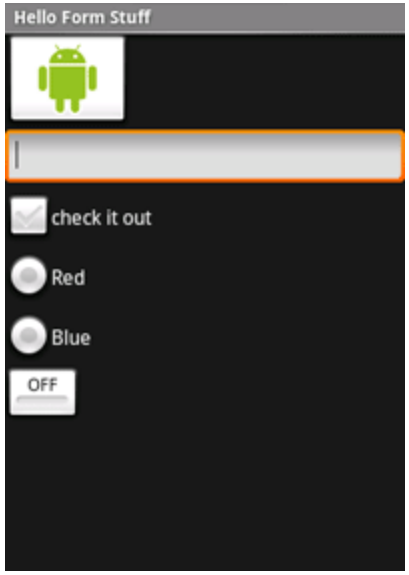
- Containers
- Resources
- Build GUI with XML
- Intents
- Manifest
- *Toast*

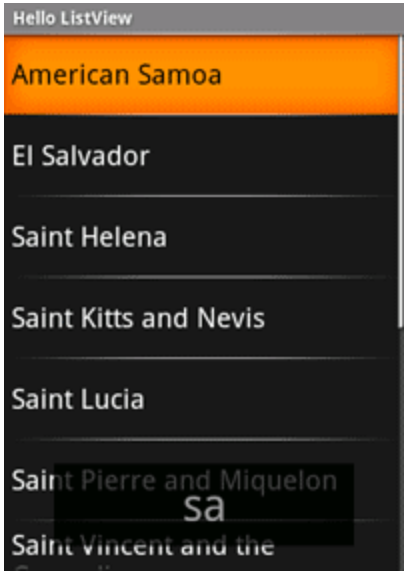
Components (Widgets)

- Extend **View**
- Placed on an Activity
  - Static
    - TextView
    - ProgressBar
    - ImageView
    - …
  - Dynamic
    - Button
    - EditText
    - CheckBox
    - RadioButton
    - SeekBar
    - Spinner
    - Gallery
    - MapView
    - ListView
    - …



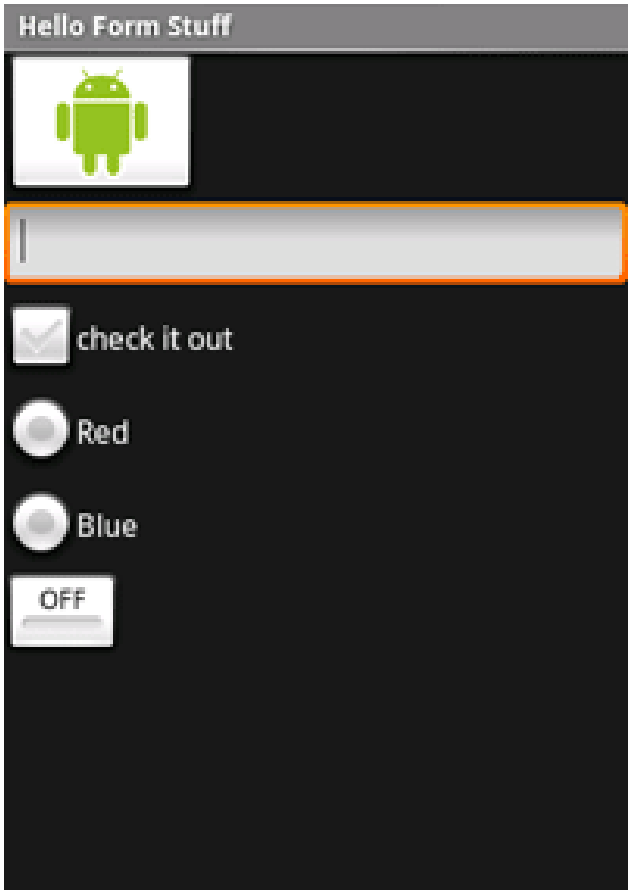






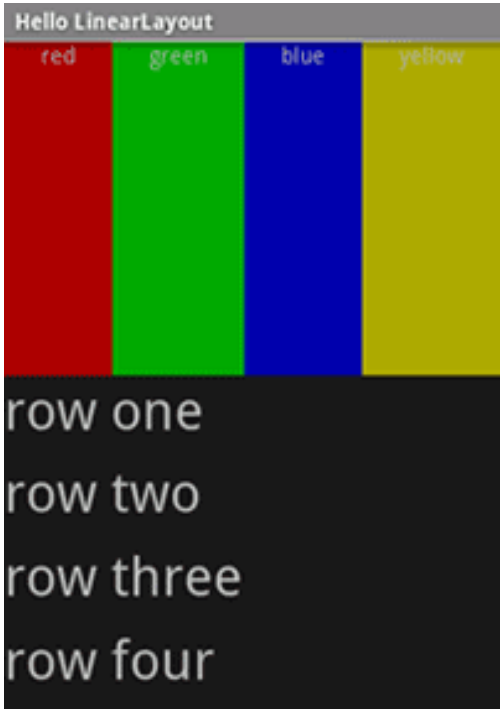
Container

- View
  - Contains other Views
  - Layout
- Types
  - LinearLayout
  - RelativeLayout
  - ScrollView
  - TableLayout
  - ConstraintLayout
- They are used combined



Container Types

LinearLayout



RelativeLayout

Container Types

PDM
sayHello();

Constraint Layout

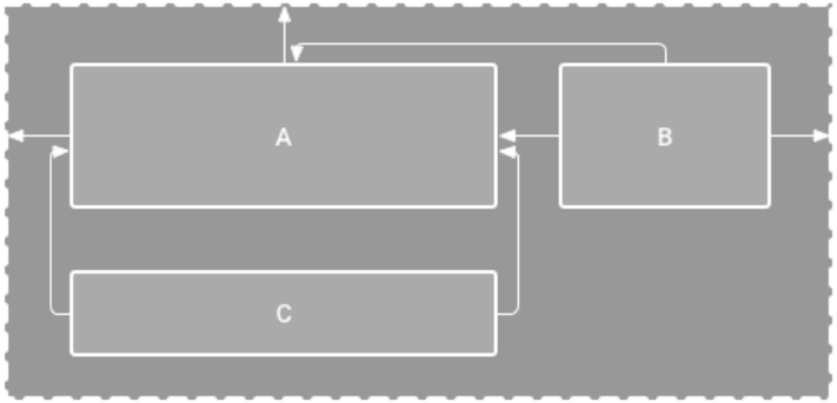


Figure 1. The editor shows view C below A, but it has no vertical constraint.

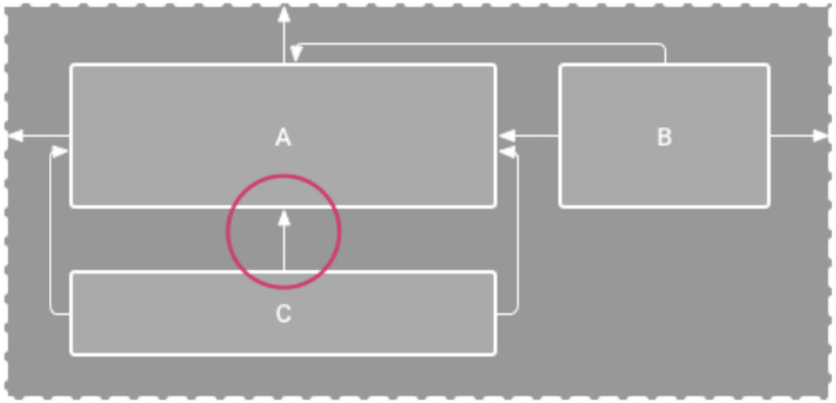


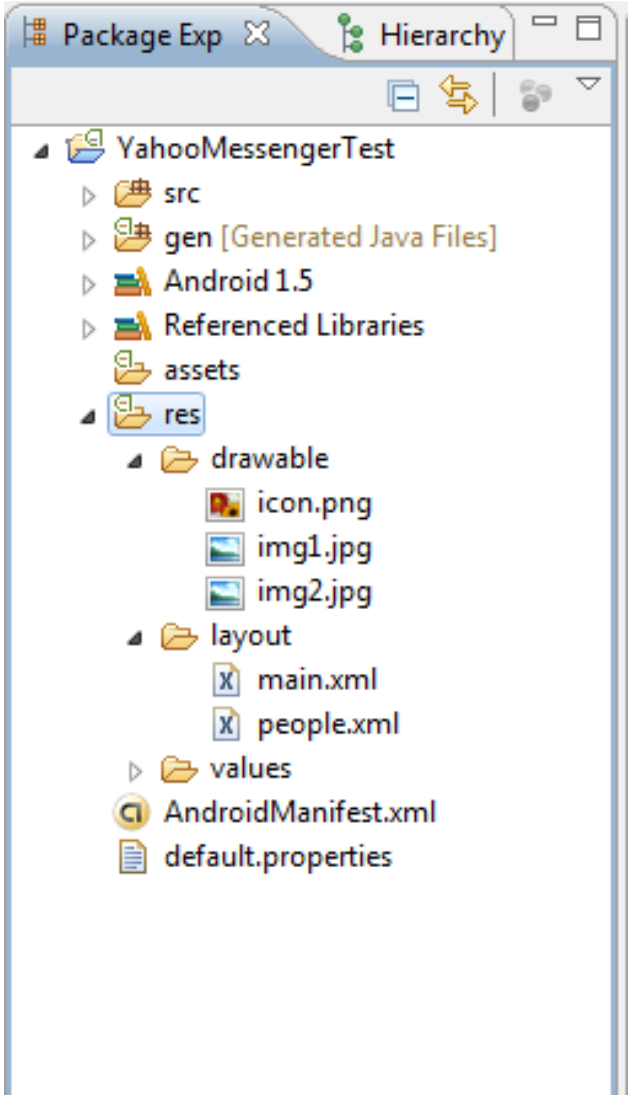
Figure 2. View C is now vertically constrained below view A.

Resources

- **res** folder (from the source)
- Images
  - *drawable -_dpi*
    - *ldpi*
    - *mdpi*
    - *hdpi*
- UI
  - *layout*
- Constants
  - *Values*
    - *strings.xml*
- Raw
  - *Unmodifiable* resources

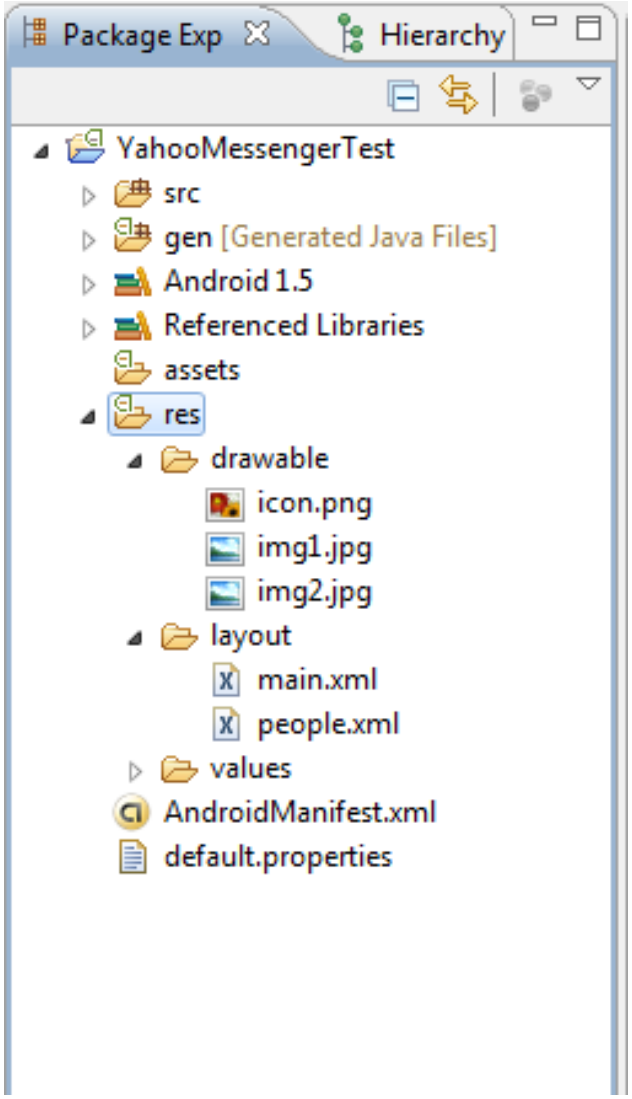
Automatic resources in Java code

- Drawable
  - Images
  - **R**.drawable.*name*
    - **R**.drawable.*icon*
    - **R**.drawable.*img1*
    - **R**.drawable.*img2*
- Layout
  - Containers
  - **R**.layout.*name*
    - **R**.layout.*main*
    - **R**.layout.*people*



Automatic resources in XML

- Drawable
  - Images
  - @drawable/name
    - @drawable/icon
    - @drawable/img1
    - @drawable/img2
- Layout
  - Containers
  - @layout/name
    - @layout/main
    - @layout/people



Constructing the GUI with XML

- Complex component
- Simple code
- No Java code
- In the Activity code
  - **this**.setContentView (**R**.layout.*name*);

```
<?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"
    >
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Hello"
        />

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello"
        />
</LinearLayout>
```

GUI XML example file


```
<?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"
    >
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Hello"
        />

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello"
        />
</LinearLayout>
```

GUI XML example file

PDM
sayHello();

- *xmlns:android*
  - Only the root component
  - mandatory
- android:layout_width
- android:layout_height
  - mandatory
  - Values
    - wrap_content
    - match_parent
    - n **px**

```
<?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"
    >
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Hello"
        />

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello"
        />
</LinearLayout>
```

GUI XML example file

- *android:id*
  - Usage of the component in the Java code
  - **@**+id/name
    - **R**.id.*name*
  - Example
    - **@**+id/button1
      - **R**.id.*button1*

```
<?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"
    >
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Hello"
        />

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello"
        />
</LinearLayout>
```

GUI XML example file

- *android:id*
  - Usage of the component in the Java code
  - **@**+id/name
    - **R**.id.*name*
  - Example
    - **@**+id/button1
      - **R**.id.*button1*

```
package com.ymsgtest;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class YahooMessengerTest extends Activity {
    /** Called when the activity is first created. */

    Button button1;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        button1 = (Button)this.findViewById(R.id.button1);
        button1.setOnClickListener(new OnClickListener()
        {
            public void onClick(View v)
            {
                button1.setText("Clicked");
            }
        });
    }
}
```

GUI XML example file

XML

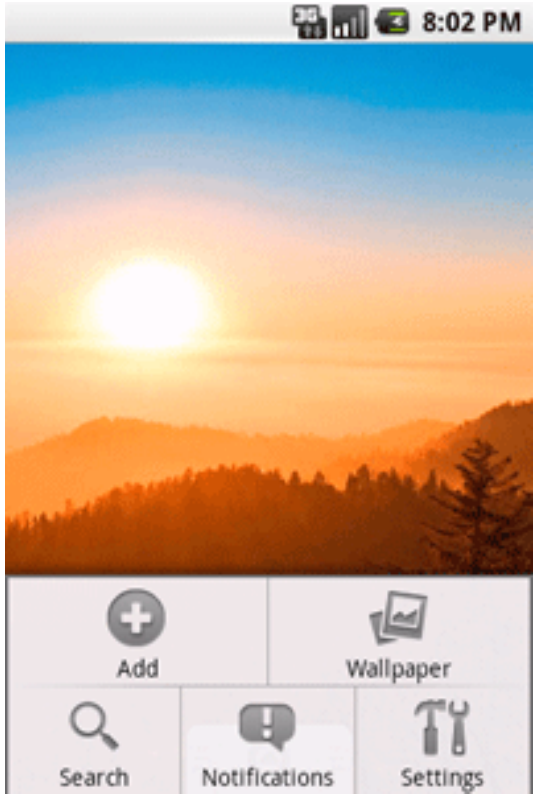
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
        <TextView
            android:layout_width="fill_parent"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

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello"
        />
</LinearLayout>
```

Java Code

```java
package com.ymsgtest;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class YahooMessengerTest extends Activity {
    /** Called when the activity is first created. */

    Button button1;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        button1 = (Button)this.findViewById(R.id.button1);
        button1.setOnClickListener(new OnClickListener()
        {
            public void onClick(View v)
            {
                button1.setText("Clicked");
            }
        });
    }
}
```

Menu

- MENU (soft) button press
- Events
  - **public boolean** *onCreateOptionsMenu* (**Menu** menu);
  - **public boolean** *onOptionsItemSelected*(**MenuItem** item);
- Adding a menu
  - menu.*add* (…);



Menu Example

```
/* Creates the menu items */
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(0, MENU_NEW_GAME, 0, "New Game");
    menu.add(0, MENU_QUIT, 0, "Quit");
    return true;
}

/* Handles item selections */
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case MENU_NEW_GAME:
        newGame();
        return true;
    case MENU_QUIT:
        quit();
        return true;
    }
    return false;
}
```

Menu using XML

XML (res/**menu**/meniu.xml**)**

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/new_game"
          android:title="New Game"
          android:icon="@drawable/img1"/>
    <item android:id="@+id/quit"
          android:title="Quit"
          android:icon="@drawable/img2"/>
</menu>
```

Java Code

```
/* Creates the menu items */
public boolean onCreateOptionsMenu(Menu menu)
{
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.meniu, menu);
    return true;
}

/* Handles item selections */
public boolean onOptionsItemSelected(MenuItem item)
{
    switch (item.getItemId())
    {
        case R.id.new_game:
            Toast.makeText(this, "New Game", Toast.LENGTH_LONG).show();
            return true;
        case R.id.quit:
            Toast.makeText(this, "Quit", Toast.LENGTH_LONG).show();
            return true;
    }
    return false;
}
```

Menu using XML

- XML (res/menu/meniu.xml)

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/new_game"
          android:title="New Game"
          android:icon="@drawable/img1"/>
    <item android:id="@+id/quit"
          android:title="Quit"
          android:icon="@drawable/img2"/>
</menu>
```

Menu using XML

- Java Code

```
/* Creates the menu items */
public boolean onCreateOptionsMenu(Menu menu)
{
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.meniu, menu);
    return true;
}

/* Handles item selections */
public boolean onOptionsItemSelected(MenuItem item)
{
    switch (item.getItemId())
    {
        case R.id.new_game:
            Toast.makeText(this, "New Game", Toast.LENGTH_LONG).show();
            return true;
        case R.id.quit:
            Toast.makeText(this, "Quit", Toast.LENGTH_LONG).show();
            return true;
    }
    return false;
}
```

Intents

- Intent
  - A component that wants something
- Generates events
  - Phone call
  - Send sms
- Object

The intent is described by

- **Action**
- **Data**
- Category
- Type
- Component
- Extras

Action

- Desired action
  - Call
  - Send sms
  - Open document
  - Edit document
  - Start an activity
- String
  - “android.intent.action.MAIN”
  - “android.intent.action.VIEW”

Action example

- ACTION_MAIN
- ACTION_VIEW
- ACTION_EDIT
- ACTION_PICK
- ACTION_CHOOSER
- ACTION_GET_CONTENT
- ACTION_DIAL
- ACTION_CALL
- ACTION_SEND
- ACTION_SENDTO
- ACTION_ANSWER
- ACTION_SEARCH
- ACTION_WEB_SEARCH
- …

Exampe of *informative* action

PDM
sayHello();

- ACTION_TIME_TICK
- ACTION_TIME_CHANGED
- ACTION_TIMEZONE_CHANGED
- ACTION_BOOT_COMPLETED
- ACTION_PACKAGE_ADDED
- ACTION_PACKAGE_CHANGED
- ACTION_PACKAGE_REMOVED
- ACTION_PACKAGE_RESTARTED
- ACTION_PACKAGE_DATA_CLEARED
- ACTION_UID_REMOVED
- ACTION_BATTERY_CHANGED
- ACTION_POWER_CONNECTED
- ACTION_POWER_DISCONNECTED
- ACTION_SHUTDOWN

Data

- Associated data
- Usually an URI
  - tel:123
  - content://contacts/…
  - File:///sdcard/...

Category

- Defines *better the* action
  - **CATEGORY_LAUNCHER**
  - CATEGORY_ALTERNATIVE
  - CATEGORY_DEFAULT
- Additional information
- Optional
- String
  - "android.intent.category.MAIN“
  - "android.intent.category.ALTERNATIVE"

Category example

- CATEGORY_DEFAULT
- CATEGORY_BROWSABLE
- CATEGORY_TAB
- CATEGORY_ALTERNATIVE
- CATEGORY_SELECTED_ALTERNATIVE
- **CATEGORY_LAUNCHER**
- CATEGORY_INFO
- *CATEGORY_HOME*
- CATEGORY_PREFERENCE
- CATEGORY_TEST
- CATEGORY_CAR_DOCK
- CATEGORY_DESK_DOCK
- CATEGORY_CAR_MODE

Type

- Data type
  - It is generally determined in
    - file:///sdcard/poze/poza.jpg
  - It has priority over
- Optional
- String
  - “image/jpg”
  - “contacts/contact”

Component

- Mentions the **specific destination component**
- Optional
- Start/Stop
  - Activities
    - *startActivity (…)*
    - *finishActivity (…)*
  - Services
    - *startService (…)*
    - *stopService (…)*

Extras

- Extra data
- Bundle
  - *putExtra ()*
  - *getStringExtra ()*
  - *getIntExtra ()*
  - *getBooleanExtra ()*
  - …

Usage

- Functions which *emit* intents
  - **void** *startActivity* (**Intent** i)
  - **void** *startService* (**Intent** i)
  - **void** *sendBroadcast* (**Intent** i, **String** permission)
- The system searches for the appropriate component
  - *Intent Resolution*

Search for the activity

- Filter arguments
  - ACTION
  - TYPE (usually extracted from DATA)
  - CATEGORY
- Determined
  - **Determines the appropriate component**
  - Determines the appropriate components list

IntentFilters

- Components that declare filters
  - Activities
  - Services
  - BoradcastReceivers
- In AndroidManifest.xml

Example


```
<activity android:name="MainActivity" android:exported="true">
    <!-- This activity is the main entry, should appear in app launcher -->
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

The applications menu asks for a list of components that respond to
Action: ACTION_MAIN
Category: CATEGORY_LAUNCHER

Exemple (2)

```
<activity android:name="ShareActivity" android:exported="false">
    <!-- This activity handles "SEND" actions with text data -->
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="text/plain"/>
    </intent-filter>
    <!-- This activity also handles "SEND" and "SEND_MULTIPLE" with media data -->
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <action android:name="android.intent.action.SEND_MULTIPLE"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="application/vnd.google.panorama360+jpg"/>
        <data android:mimeType="image/*"/>
        <data android:mimeType="video/*"/>
    </intent-filter>
</activity>
```

Exemple (3)

PDM
sayHello();

```
<activity class=".NotesList" android:label="@string/title_notes_list">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <action android:name="android.intent.action.EDIT" />
        <action android:name="android.intent.action.PICK" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.GET_CONTENT" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
    </intent-filter>
</activity>

<activity class=".NoteEditor" android:label="@string/title_note">
    <intent-filter android:label="@string/resolve_edit">
        <action android:name="android.intent.action.VIEW" />
        <action android:name="android.intent.action.EDIT" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
    </intent-filter>

    <intent-filter>
        <action android:name="android.intent.action.INSERT" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
    </intent-filter>

</activity>
```

Manifest

AndroidManifest.xml

- Declare components (Activities, Services, etc)
- Declare permissions

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.ymsgtest"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="Yahoo Messenger Test">
        <activity android:name=".YahooMessengerTest"
                  android:label="Yahoo Messenger Test Activity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>
    <uses-sdk android:minSdkVersion="3" />

    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
</manifest>
```

Starting other components

- Starting Activities

```
// Create the text message with a string.
val sendIntent = Intent().apply {
    action = Intent.ACTION_SEND
    putExtra(Intent.EXTRA_TEXT, textMessage)
    type = "text/plain"
}

// Try to invoke the intent.
try {
    startActivity(sendIntent)
} catch (e: ActivityNotFoundException) {
    // Define what your app should do if no activity can handle the intent.
}
```

Starting other components

- Starting Services

```
// Executed in an Activity, so 'this' is the Context
// The fileUrl is a string URL, such as "http://www.example.com/image.png"
val downloadIntent = Intent(this, DownloadService::class.java).apply {
    data = Uri.parse(fileUrl)
}
startService(downloadIntent)
```

Starting other components

- Sending broadcasts

```
Intent().also { intent ->
    intent.setAction("com.example.broadcast.MY_NOTIFICATION")
    intent.putExtra("data", "Nothing to see here, move along.")
    sendBroadcast(intent)
}
```

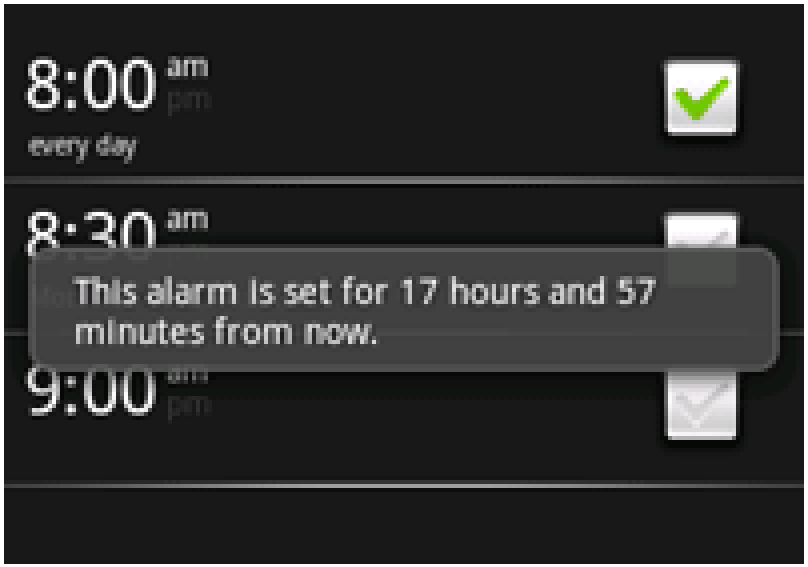
Stopping an activity

- context.*finishActivity* (**Intent** intent);
  - Another component wants to stop the Activity
- *finish* ();
  - The Activity stops itself

```
Intent starter = new Intent (context,
   ActivityClass.class);
Context.finishActivity (starter);
```

Toast

- Display notifications
  - Short
    - Toast.LENGTH_SHORT
  - Long
    - Toast.LENGTH_LONG
- **Toast**




```
Toast.makeText(this, "New Game", Toast.LENGTH_LONG).show();
```

Conclusions

- Task
  - A stack of activities
- Intents

- Manifest
  - Declare activities
- Automatic Resources
  - Drawable
    - **R**.drawable.*nume*
    - **@**/drawable/*nume*
  - Layout
    - Fișiere XML
    - **R**.layout.*nume*
    - **@**/layout/*nume*
- Toast
  - Display messages

Keywords

- Task
  - Stack
- Activity
  - Lifecycle
- Containers
  - Layouts
  - LinearLayout
  - TableLayout
  - RelativeLayout
  - GridLayout
- Toast
- XML GUI
- Menu
- Manifest

- Components
  - Static
    - TextView
    - ProgressBar
    - ImageView
    - …
  - Dynamic
    - Button
    - EditText
    - CheckBox
    - RadioButton
    - SeekBar
    - Spinner
    - Gallery
    - MapView
    - ListView

Quenstions