

Chapter 5 – Android Resources

[Group 7]

Chapter Objective

- To know what is Android Resources and kinds of resources.
- Understand briefly how each resources works

What is Android Resources ?

- Resources: bundled to the app, stored in res/directory, can be accessed by code line:
R.<category>.<resourceName>.
- Content: layouts, values, drawables, raw, styles, design guildelines.

Layout

• Definition

- A way to organize Views inside an UI, can be created in XML files in res/layout, nested.

• Layout XML

- Containers (ViewGroups) contain Views (TextView, ImageView, EditText, Button, ImageButton,...).
- Required: layout_width, layout_height. Optional: id (findViewById()).

• Adaptive: Layout on Android

- Use different layout XMLs in different directories. Eg: tablet: layout-large, layout-xlarge, Phone: layout-normal,...
- To load XML layout: In Activity file, use onCreate() with setContentView();
Fragment: use onCreateView().

• Popular Layout classes: FrameLayout, LinearLayout, RelativeLayout, ViewPager.

- FrameLayout: can contain multiple children, multiple layers, first child will be at the bottom, support child margins, gravity.
- LinearLayout: one direction, horizontal or vertical.
- RelativeLayout: enables to specify how child views are positioned “relative” to each other.

View Pager

- **Create Viewpager**

- Horizontal swipe gesture, page by page scrolling.
- Each page tab is a fragment.
- How to create ViewPager ?

+ XML layout, add a short code

```
<android.support.v4.view.ViewPager
    android:id="@+id/pager"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

+ ViewPager accessed through “Adapter” in Java class.

+ Setup ViewPager in Activity’s onCreate().

- **Create Header for ViewPager:** use TabLayout

- In XML file: add a short code

```
<android.support.design.widget.TabLayout
    android:id="@+id/tab"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

- In Activity/Fragment: setup a link

```
TabLayout tabLayout = (TabLayout) findViewById(R.id.tab);
tabLayout.setupWithViewPager(pager);
```

Values

- **Types of values**

- String: res/string.xml

```
<string name="refresh">Refresh</string>
```

- Integers: res/integers.xml

```
<integer name="column_count">1</integer>
```

- Booleans: res/bool.xml

```
<bool name="is_tablet">>false</bool>
```

- Colors: res/colors.xml

```
<color name="colorPrimary">#3F51B5</color>
```

- Sizes: res/dimens.xml

- ```
<dimen name="title_width">60dp</dimen>
<dimen name="title_height">24dp</dimen>
<dimen name="title_font_size">48sp</dimen>
```
- Arrays: res/arrays.xml
  - Themes/ Styles: res/styles.xml

## Drawables

- 2 types:
  - + XML drawables: res/drawable.
  - + Bitmap drawable (PNG/JPEG): res/drawable-\*dpi.
- ImageView: src="drawable/name".
- View: background="drawable/name".
- 9patch Drawables:** \*.9.png, transparent, stretch certain part of the image, padding.

**Raw Data:** embedded in apps, fonts, JSONs, audio, HTML/CSS/JS, etc.

- Place to store raw: res/raw or assets/.
- Raw is accessible with Context.getResources().

\*How to use Raw data

- Use input stream to access binary data

```
InputStream is = context.getResources()
 .openRawResource(R.raw.resid);
```

- Use file name with Context.getAssets()

```
InputStream is = getAssets().open("<filename>");
```

## Menu

- **App Bar:** Navigation Drawer, App logo, Activity Title, Actions.

How to use App Bar

- Add Toolbar to the Activity layout.

```
<android.support.v7.widget.Toolbar
android:id="@+id/my_toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="?attr/colorPrimary"
android:elevation="4dp"
```

```
android:theme="@style/ThemeOverlay.AppCompat.ActionBar"
app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>
```

- Setup in onCreate(): add code

```
Toolbar myToolbar = (Toolbar) findViewById(R.id.my_toolbar);
setSupportActionBar(myToolbar);
```

- Define menu resources: defined in res/menu/weather\_menu.xml.

- Inflate the menu:

```
public class MainActivity extends AppCompatActivity {
 ...
 @Override
 public boolean onCreateOptionsMenu(Menu menu) {
 getMenuInflater().inflate(R.menu.menu_weather, menu);
 return true;
 }
}
```

- Response to actions: override onOptionsItemSelected()

```
public class MainActivity extends AppCompatActivity {
 ...
 @Override
 public boolean onOptionsItemSelected(MenuItem item) {
 switch (item.getItemId()) {
 case R.id.search:
 return true;
 default:
 super.onOptionsItemSelected(item);
 }
 }
}
```