Advanced Android Widgets

Agenda

- Android App Widgets
- App Scroll View
- App View Pager
- App Action Bar
- App Custom View
- App Progress Bar

Widgets in Android

- Widget is a small control placed on Home Screen for quick access
- It allows to put favorite app on home scree for always available.
- Some popular and mostly used widgets includes weather, fitness, Clock, etc
- Widgets could be of many types such as
    - ***Widgets for Information Statistics***
    - ***Widgets for Live data collection***
    - ***Widgets to Control Mobile Actions***

Widgets - XML File

The steps for creating a widget are:-

- AppWidgetProviderInfo object will be created and a separate XML file will be created for each widget.
- Inside your project a folder need to be created named as xml.
- Create an XML file for the widget
- The resource type should be set to **AppWidgetProvider** in XML**.**

Widgets - XML File

XML file code sample is give below:-

```
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="146dp"
    android:updatePeriodMillis="0"
    android:minHeight="146dp"
    android:initialLayout="@layout/activity_main">
</appwidget-provider>
```

- *Widget - Layout file*
- *Widget - Java file*

Widgets Methods

Some important methods used for Widgets are given below:

- **onDeleted(Context context, int[] appWidgetIds)**
- **onDisabled(Context context)**
- **onEnabled(Context context)**
- **onReceive(Context context, Intent intent)**

Widget - Manifest File

```
<receiver android:name="ExampleAppWidgetProvider" >
   <intent-filter>
      <action
android:name="android.appwidget.action.APPWIDGET_UPDATE" />
   </intent-filter>
   <meta-data android:name="android.appwidget.provider"
      android:resource="@xml/example_appwidget_info" />
</receiver>
```

Android App Scroll View

- ScrollView is used to provide a vertical scrollebale view for view
- A single direct child is used in ScrollView
- Multiple views can be created using other Layout like Linear Layout
    - By default, Only vertical scrolling is allowed in ScrollView
    - Specifically, HorizontalScrollView is used to create a horizontally scrollable view.

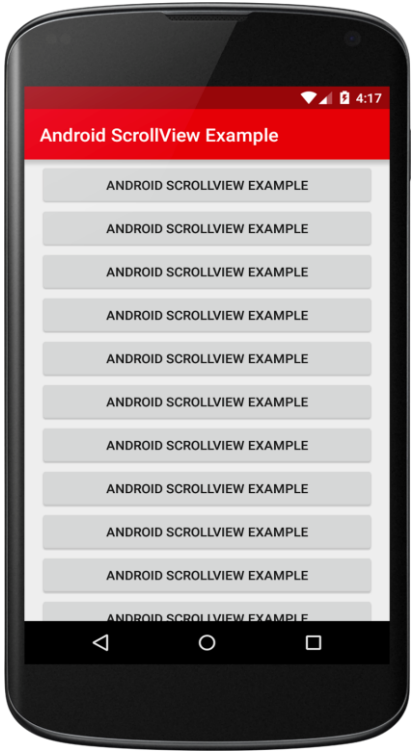
Android Scroll View

XML attributes of used for ScrollView

Attribute	Description
fillViewport	Used to define the filling of view point.

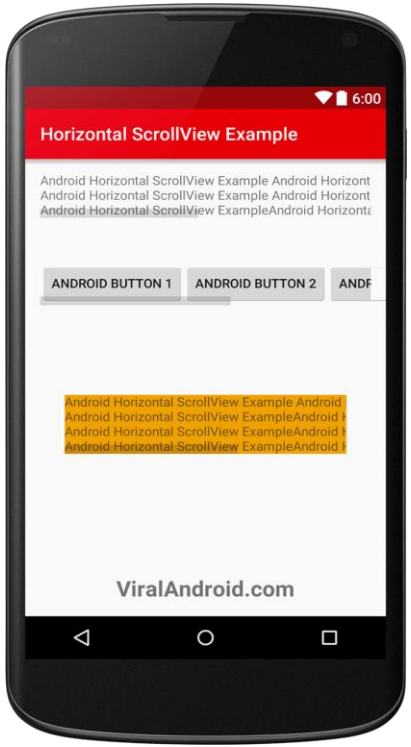
Android Scroll View (Vertical)

- **android.widget.ScrollView** class is used for Scollview functionality.
- Within the Vertical ScrollView's palette, the child items can be vertically scrolled.



Android Scroll View (Horizontal)

- It is a Frame Layout used by **HorizontalScrollView**
- **android.widget.HorizontalScrollView** class provides the capability of a horizontal scroll view and enables horizontal scrolling of the child items or views.

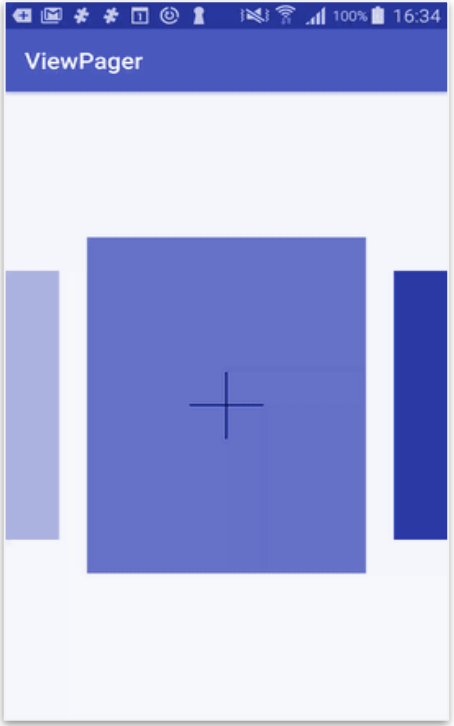


Page Viewer in Android App

- A layout manager called ViewPager enables users to navigate across data pages by swiping left and right.
- The user typically shifts right to left to navigate to another screen in apps like Snapchat and YouTube.
- A View Pager switches views by using fragments rather than activities.

ViewPager in Android App

- The main layout in an XML layout is normally where the ViewPager widget is added.
- Extending the FragmentPagerAdapter or FragmentStatePagerAdapter class to create an adapter.
- The Viewpager's pages are filled out by an adaptor.
- FragmentPagerAdapter and FragmentStatePagerAdapter extend the base class PagerAdapter.



Action Bar in Android App

- The component located at the top of the activity screen is called the ActionBar.
- A mobile application's uniform presence across all of its functions is a key component.
- It gives the programme a visual framework and includes some of the components that users utilise most frequently.

Action Bar in Android

ActionBar has four major components listed below:

- **App Icon**:
- **View Control**:
- **Action Buttons**:
- **Action Overflow**:

Android Custom View

- Based on the fundamental layout classes, Android provides a comprehensive and potent componentized paradigm for creating your user interface: View and ViewGroup
- To start, you can create your UI using a selection of prebuilt View and ViewGroup subclasses on the platform, which are referred to as widgets and layouts, respectively.

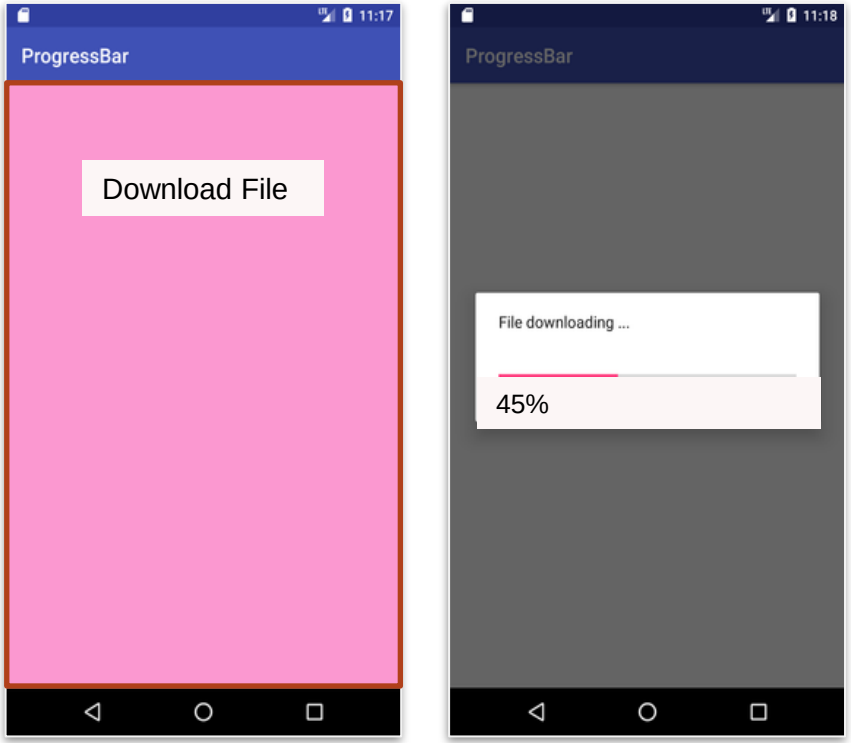
Android App Progress Bar

- To display the progress of work being done, such as downloading files or assessing work status, we can use the Android progress bar dialogue box.
- **ProgressDialog** class provides following methods
  - **setProgress()**
  - **setMessage()**
  - **setProgressStyle()**
  - **setMax(), show()**
  - **etc**
- The Progress Dialog has progress bar range 0 to 10000

Android App Progress Bar (Coding Example)

```
ProgressDialog progressBar = new ProgressDialog(this);

progressBar.setCancelable(true);//you can cancel it by pressing back button

progressBar.setMessage("File downloading ...");

progressBar.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);

progressBar.setProgress(0);//initially progress is 0

progressBar.setMax(100);//sets the maximum value 100

progressBar.show();//displays the progress bar
```

Android Progress Bar



Questions