<http://www.tutorialspoint.com/android/index.htm>

1. **Application Components**
   1. **Four Main Components**
      1. **Activities**

They dictate the UI and handle the user interaction to the smart phone screen. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched.

* + 1. **Services**

They handle background processing associated with an application. A service is a component that runs in the background to perform long-running operations.

* + 1. **Broadcast Receivers**

They handle communication between Android OS and applications. Broadcast Receivers simply respond to broadcast messages from other applications or from the system. Each message is broadcaster as an Intent object

* + 1. **Content Providers**

They handle data and database management issues. A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolver class. The data may be stored in the file system, the database or somewhere else entirely.

* 1. **Other Components**
     1. **Fragments**

Represents a portion of user interface in an Activity.

* + 1. **Views**

UI elements that are drawn on-screen including buttons, lists forms etc.

* + 1. **Layouts**

View hierarchies that control screen format and appearance of the views.

* + 1. **Intents**

Messages wiring components together.

* + 1. **Resources**

External elements, such as strings, constants and drawable pictures.

* + 1. **Manifest**

Configuration file for the application. Whatever component you develop as a part of your application, you must declare all its components in a manifest.xml which resides at the root of the application project directory. This file works as an interface between Android OS and your application, so if you do not declare your component in this file, then it will not be considered by the OS.

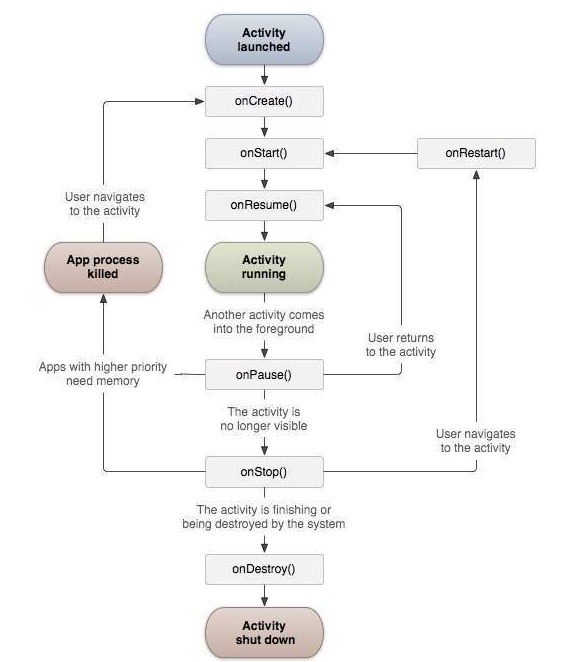
1. **Resource**
   1. **Directory**
      1. **anim/**
      2. **drawable/**
      3. **layout/**
      4. **menu/**
      5. **raw/**
      6. **values/**
      7. **xml/**
      8. **color/**
   2. **Alternative Resources**

The form of name is <resources\_name>-<config\_qualifier>/, Like drawable-hdpi/.

Your application should provide alternative resources to support specific device configurations.

1. **Activitys**

Activity like a window in windows programming.

****

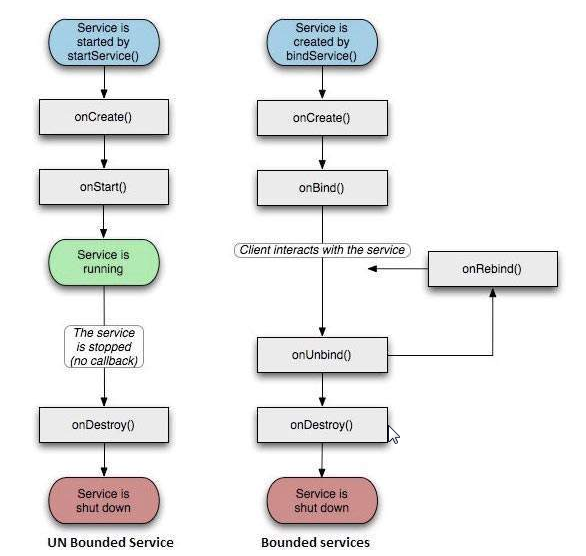
1. **Services**

A service is a component that runs in the background to perform long-running operations without needing to interact with the user and it works even if application is destroyed. A service can essentially take two states – Started and Bound.

Started: A service is started when an application component, such as an activity, starts it by calling startService(). Once started, a service can run in the background indefinitely, even if the component that started it is destroyed.

Bound: A service is bound when an application component binds to it by calling bindService(). A bound service offers a client-server interface that allows components to interact with the service, send requests, get results, and even do so across processes with interprocess communication (IPC).

A service has life cycle callback methods that you can implement to monitor changes in the service's state and you can perform work at the appropriate stage.



1. **Broadcast Receivers**

Broadcast Receivers simply respond to broadcast messages from other applications or from the system itself. These messages are sometime called events or intents. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action.

A broadcast receiver is implemented as a subclass of BroadcastReceiver class and overriding the onReceive() method where each message is received as a Intent object parameter.

An application listens for specific broadcast intents by registering a broadcast receiver in AndroidManifest.xml file.

If you want your application itself should generate and send custom intents then you will have to create and send those intents by using the sendBroadcast() method inside your activity class.

1. **Content Providers**

A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolver class. A content provider can use different ways to store its data and the data can be stored in a database, in files, or even over a network.

To query a content provider, you specify the query string in the form of a URI which has following format:

<prefix>://<authority>/<data\_type>/<id>