



TechLoop

Stay in the Loop.

Hello Android !!!

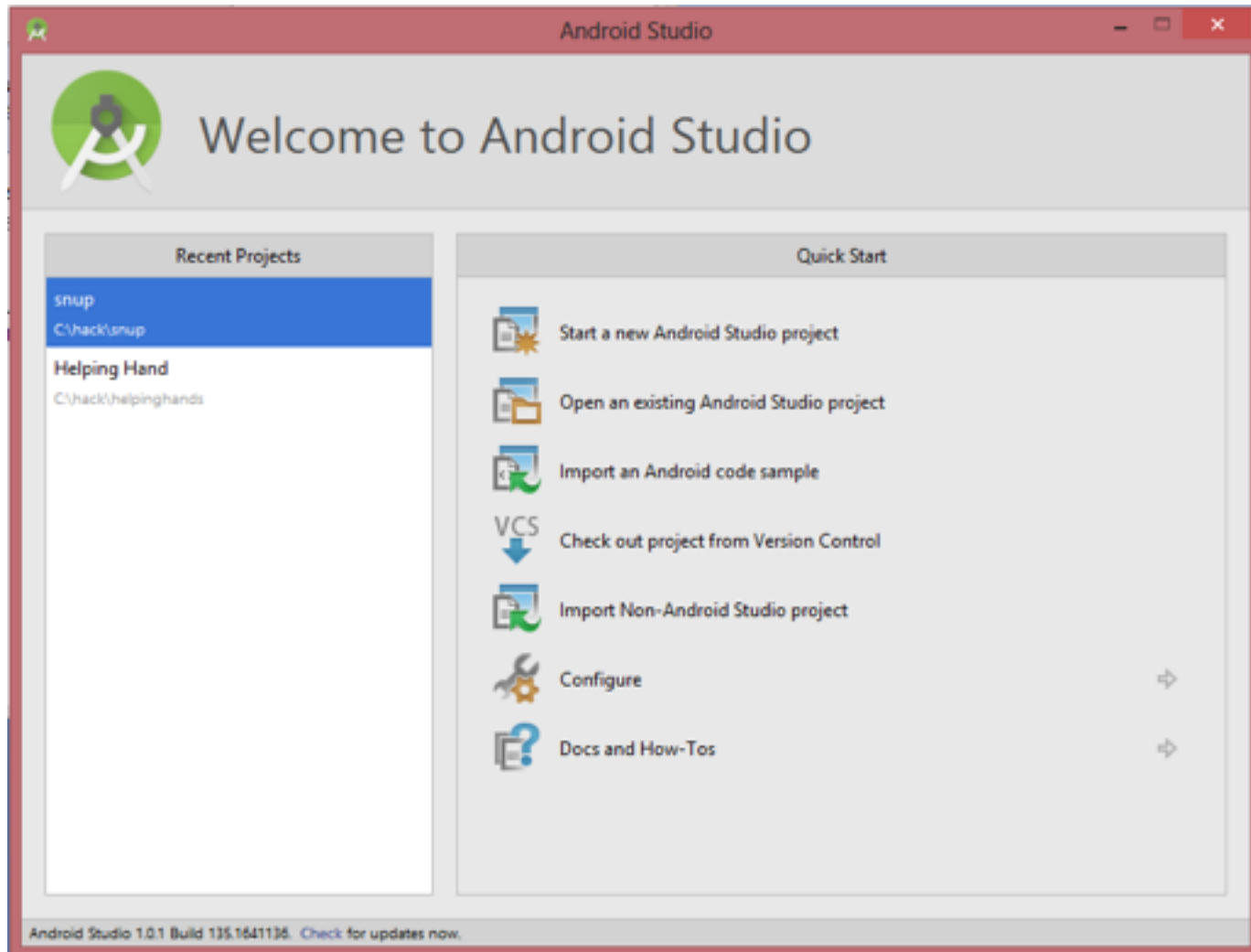


Hello Android



1. Create a new project by clicking “Start a new Project”
 - Application Name** – Hello World
 - Project Name** – HelloWorld
 - Build Target** – Android 4.3 (API)
 - Package name** – com.techloop.helloworld
 - Activity name** – MainActivity
2. Click to finish, and wait for the project to be created.
3. The build your project.
4. Select the AVD corresponding to the build target, and then wait for the emulator to load.







Create New Project

New Project

Android Studio

Configure your new project

Application name:

Company Domain:


Package name: [Edit](#)

Project location:





Create New Project

 **New Project**
Android Studio

Select the form factors your app will run on

Different platforms require separate SDKs

☒ Phone and Tablet

Minimum SDK: API 8: Android 2.2 (Froyo)

Lower API levels target more devices, but have fewer features available. By targeting API 8 and later, your app will run on approximately **100.0%** of the devices that are active on the Google Play Store. [Help me choose.](#)

☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Glass (Not Installed)

Minimum SDK:

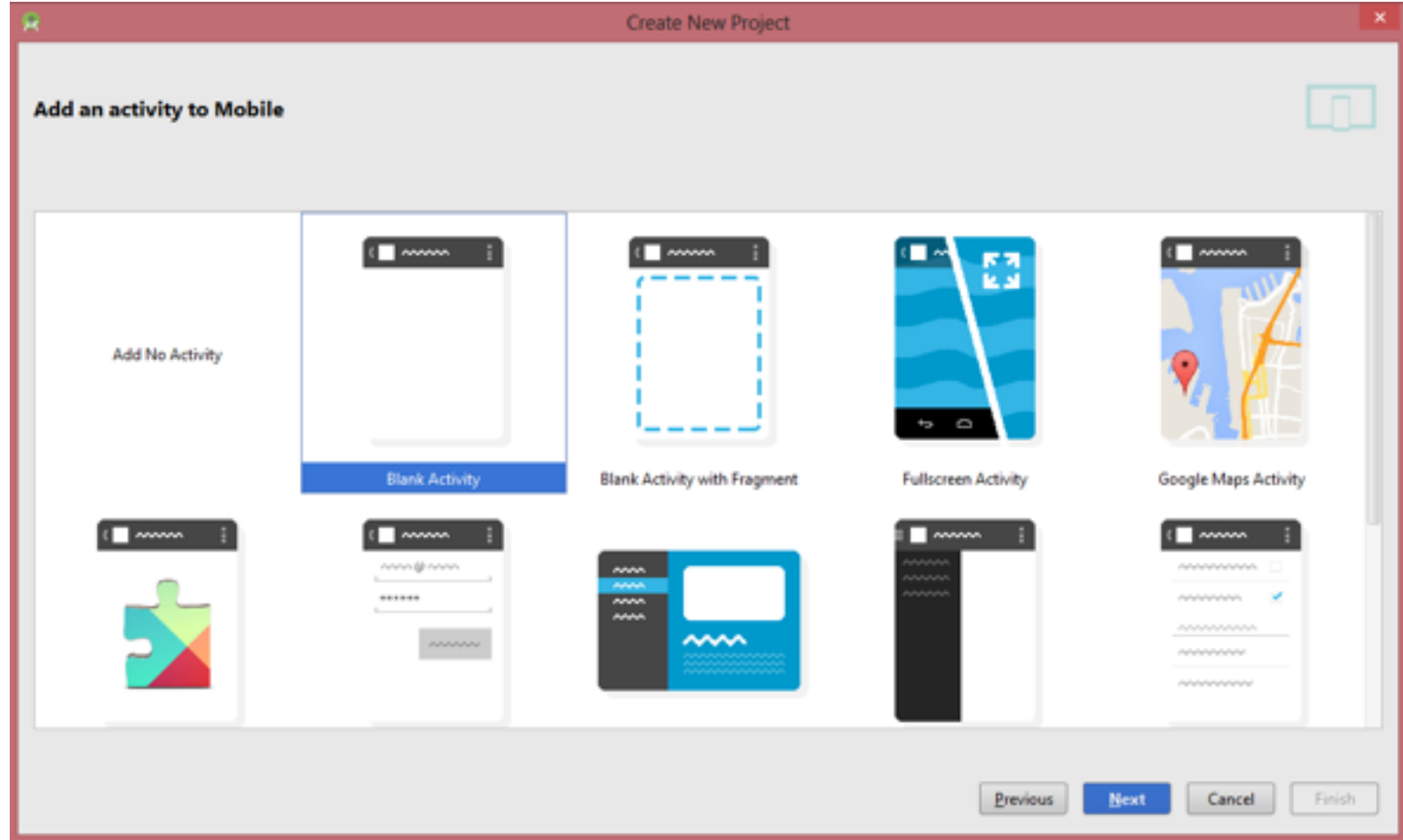
Previous

Next



Cancel


Finish








 Create New Project 

Choose options for your new file 



Blank Activity

Creates a new blank activity with an action bar.

Activity Name:

activity_main

Layout Name:

activity_main

Title:

MainActivity

Menu Resource Name:

menu_main

The name of the activity class to create

Previous

Next

Cancel

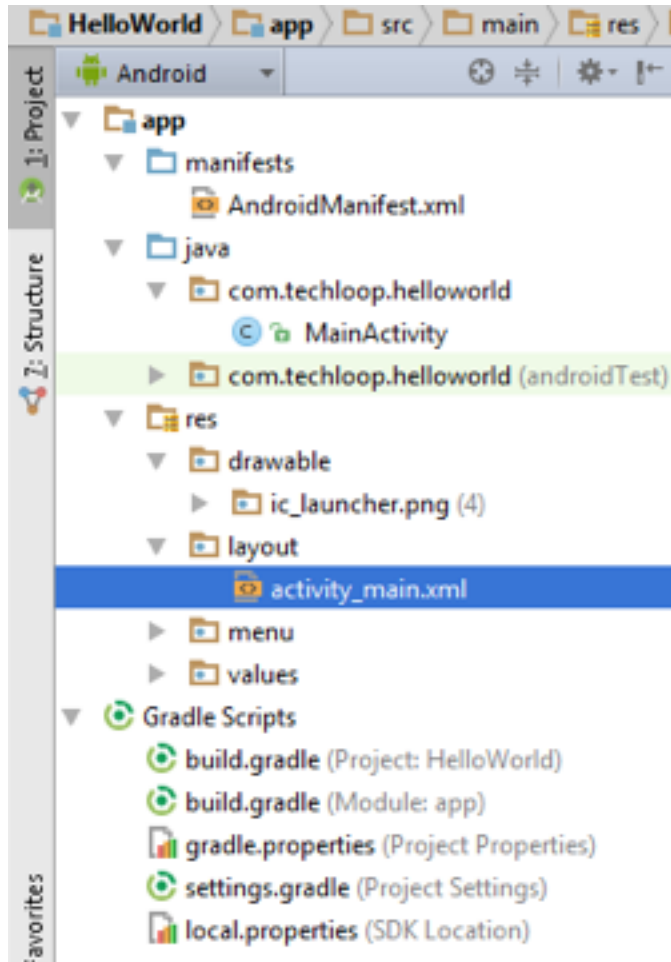
Finish



Understanding Hello world



Android Project file structure



Project Name

1.manifests

-AndroidManifest.xml

2.Java

-MainActivity

3.Res

drawable

- ic_launcher.png

layout

-activity_main.xml

menu

values

6.Gradle Scripts



Brief Description about the Folders



Java folder

This java folder, contains the actual source code written in java programming language

Resources folder (res)

The Resource aka res folder, contains all the resource files which are divided into generally 4 folders

- a. *drawables*
- b. *layout*
- c. *menu*
- d. *values*

Apart from all other files are added to the assets folder like java projects,



AndroidManifest.xml



The manifest lets you define the structure and metadata of your application, its components, and its requirements.

Breaking down

1. **Activity tag** – To describe any new activity in the application
2. **Application, intent-filter tag**

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="Hello World"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:label="Hello World" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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





3. Android Permissions (uses-application tag)

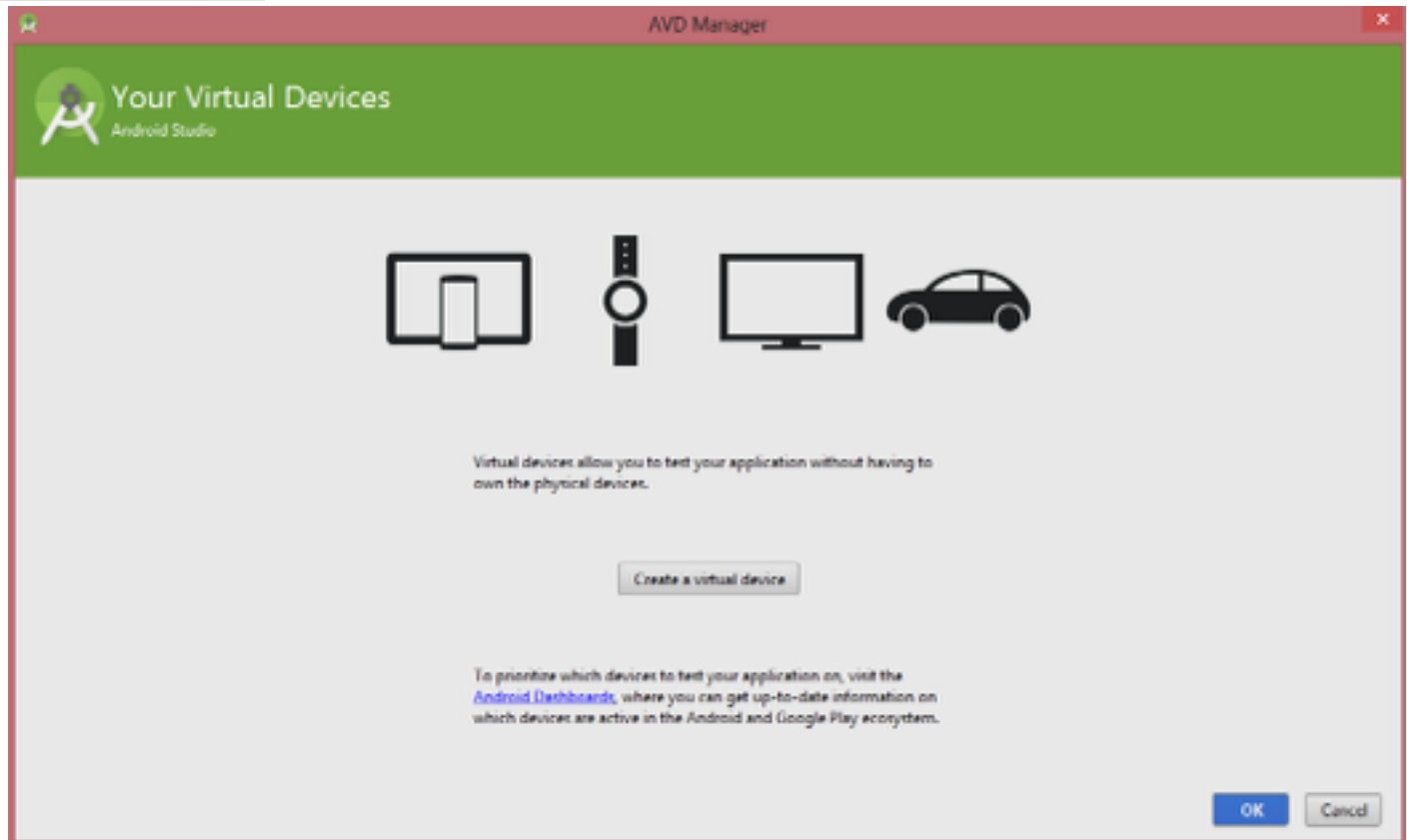
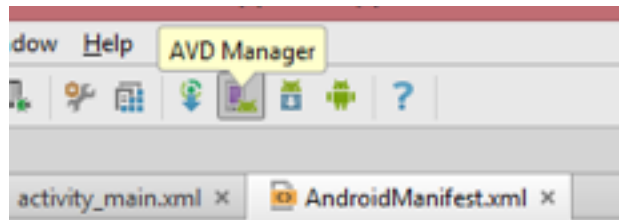
The Android permissions are the set of hardware/software permissions to be taken by the current application.

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

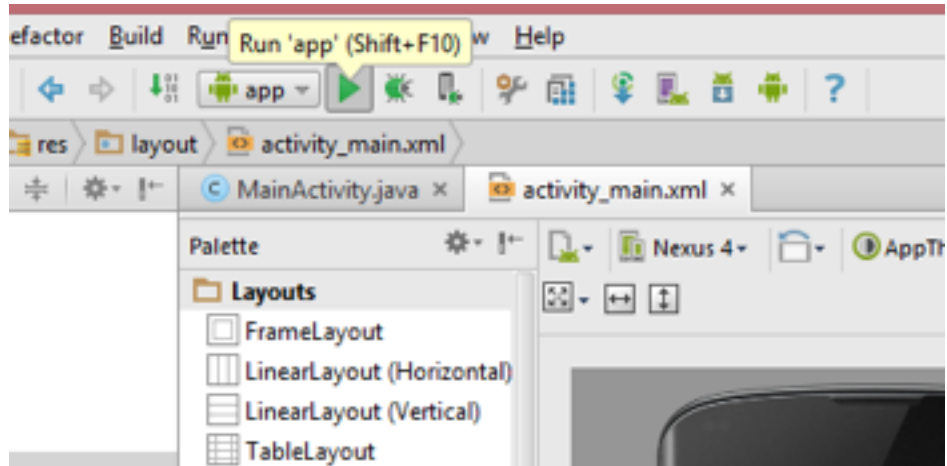
```
//  
<activity android:name=".TnCWebView"  
    android:noHistory="true"/>
```

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





Let's test our app!



Android Activity



An Activity is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map.

The very first step in developing an Android application is making a Activity
Your application must have at least one Activity.



Importance of Activity



An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI .

While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows or embedded inside of another activity (using ActivityGroup).



Creating an Activity



To create an activity, you must create a subclass of **Activity** (or an existing subclass of it). In your subclass, you need to implement the most important callback methods are:

onCreate()

The system calls this when creating your activity.

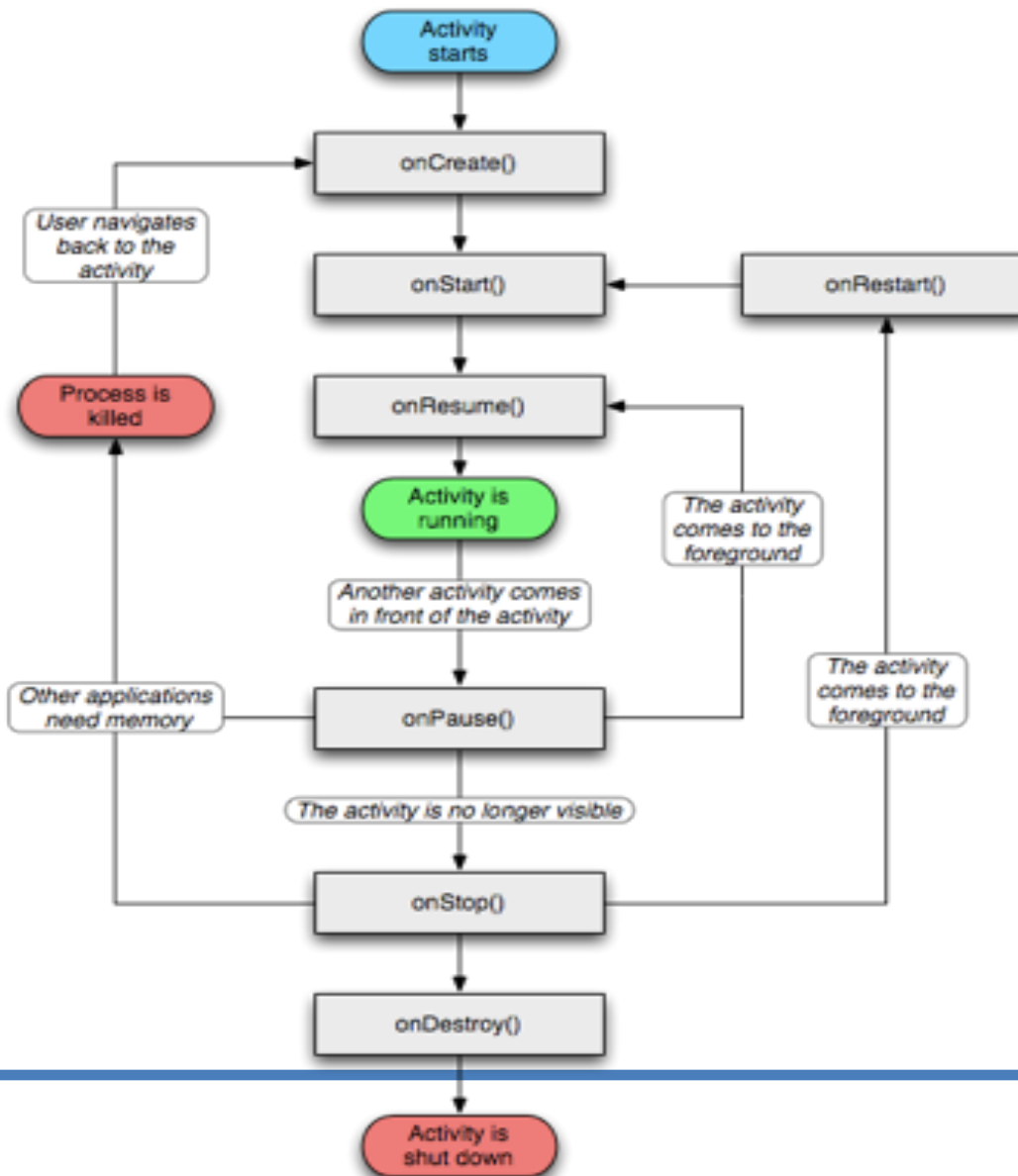
Most importantly, this is where you must call **setContentView()** to define the layout for the activity's user interface.

onPause()

This is usually where we should commit any changes that should be persisted beyond the current user session (because the user might not come back).



Activity Life Cycle



Android Service



A Service is an application component that can perform long-running operations in the background and does not provide a user interface.

Additionally, a component can bind to a service to interact with it and even perform interprocess communication (IPC).



References



1. <http://developer.android.com/reference/android/app/Activity.html>
2. <http://developer.android.com/reference/android/app/Service.html>



