MODULE - 4

Android - Developer Tools

The android developer tools let us to create interactive and powerful application for android platform. The tools can be generally categorized into two types.

- SDK tools
- Platform tools

1. SDK tools

SDK tools are generally platform independent and are required no matter which android platform we are working on. When you install the Android SDK into our system, these tools get automatically installed. The list of SDK tools has been given below −

Sl.No	Tool & description
1	**Android** This tool let us to manage AVDs, projects, and the installed components of the SDK
2	**Ddms** This tool let us to debug Android applications
3	**Draw 9-Patch** This tool allows us to easily create a NinePatch graphic using a WYSIWYG editor
4	**Emulator** This tools let us to test your applications without using a physical device
5	**Mksdcard** Helps us to create a disk image (external sdcard storage) that you can use with the emulator
6	**Proguard** Shrinks, optimizes, and obfuscates our code by removing unused code

7	**sqlite3** Lets us access the SQLite data files created and used by Android applications
8	**Traceview** Provides a graphical viewer for execution logs saved by our application
9	**Adb** Android Debug Bridge (adb) is a versatile command line tool that lets us to communicate with an emulator instance or connected Android-powered device.

We will discuss three important tools here that are android,ddms and sqlite3.

1.1 Android

Android is a development tool that lets us to perform these tasks:

- Manage Android Virtual Devices (AVD)
- Create and update Android projects
- Update your sdk with new platform add-ons and documentation android [global options] action [action options]

1.2 DDMS

DDMS stands for Dalvik debug monitor server, that provide many services on the device. The service could include message formation, call spoofing, capturing screenshot, exploring internal threads and file systems e.t.c

Running DDMS

From Android studio click on **Tools>Android>Android device Monitor**.

How it works

In android, each application runs in its own process and each process run in the virtual machine. Each VM exposes a unique port, that a debugger can attach to.

When DDMS starts, it connects to adb. When a device is connected, a VM monitoring service is created between adb and DDMS, which notifies DDMS when a VM on the device is started or terminated.

1.3 Sqlite3

Sqlite3 is a command line program which is used to manage the SQLite databases created by Android applications. The tool also allow us to execute the SQL statements on the fly.

There are two way through which you can use SQlite , either from remote shell or you can use locally.

Use Sqlite3 from a remote shell.

Enter a remote shell by entering the following command −

```
adb [-d|-e|-s {<serialNumber>}] shell
```

From a remote shell, start the sqlite3 tool by entering the following command −

```
sqlite3
```

Once you invoke sqlite3, you can issue sqlite3 commands in the shell. To exit and return to the adb remote shell, enter exit or press CTRL+D.

Using Sqlite3 directly

Copy a database file from your device to your host machine.

adb pull <database-file-on-device>

Start the sqlite3 tool from the /tools directory, specifying the database file −

sqlite3 <database-file-on-host>

2. Platform tools

The platform tools are customized to support the features of the latest android platform.

The platform tools are typically updated every time you install a new SDK platform. Each update of the platform tools is backward compatible with older platforms.

Some of the platform tools are listd below −

- Android Debug bridge (ADB)
- Android Interface definition language (AIDL)
- aapt, dexdump , and dex e.t.c

Android - SQLite Database

SQLite is a opensource SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation.

SQLite supports all the relational database features. In order to access this database, you don't need to establish any kind of connections for it like JDBC,ODBC e.t.c

Database - Package

The main package is android.database.sqlite that contains the classes to manage your own databases

Database - Creation

In order to create a database you just need to call this method openOrCreateDatabase with your database name and mode as a parameter. It returns an instance of SQLite database which you have to receive in your own object.Its syntax is given below

SQLiteDatabase mydatabase = openOrCreateDatabase("your database name",MODE_PRIVATE,null);

Apart from this , there are other functions available in the database package , that does this job. They are listed below

Sr.No	Method & Description
1	**openDatabase(String path, SQLiteDatabase.CursorFactory factory, int flags, DatabaseErrorHandler errorHandler)** This method only opens the existing database with the appropriate flag mode. The common flags mode could be OPEN_READWRITE OPEN_READONLY
2	**openDatabase(String path, SQLiteDatabase.CursorFactory factory, int flags)** It is similar to the above method as it also opens the existing database but it does not define any handler to handle the errors of databases
3	**openOrCreateDatabase(String path, SQLiteDatabase.CursorFactory factory)** It not only opens but create the database if it not exists. This method is equivalent to openDatabase method.
4	**openOrCreateDatabase(File file, SQLiteDatabase.CursorFactory factory)** This method is similar to above method but it takes the File object as a path rather then a string. It is equivalent to file.getPath()

Database - Insertion

we can create table or insert data into table using execSQL method defined in SQLiteDatabase class. Its syntax is given below

mydatabase.execSQL("CREATE TABLE IF NOT EXISTS TutorialsPoint(Username VARCHAR,Password VARCHAR);");
mydatabase.execSQL("INSERT INTO TutorialsPoint VALUES('admin','admin');");

This will insert some values into our table in our database. Another method that also does the same job but take some additional parameter is given below

Sl.No	Method & Description
1	**execSQL(String sql, Object[] bindArgs)** This method not only insert data , but also used to update or modify already existing data in database using bind arguments

Database - Fetching

We can retrieve anything from database using an object of the Cursor class. We will call a method of this class called rawQuery and it will return a resultset with the cursor pointing to the table. We can move the cursor forward and retrieve the data.

```
Cursor resultSet = mydatbase.rawQuery("Select * from TutorialsPoint",null);
resultSet.moveToFirst();
String username = resultSet.getString(0);
String password = resultSet.getString(1);
```

There are other functions available in the Cursor class that allows us to effectively retrieve the data. That includes

Sl.No	Method & Description
1	**getColumnCount()**: This method return the total number of columns of the table.
2	**getColumnIndex(String columnName)**: This method returns the index number of a column by specifying the name of the column
3	**getColumnName(int columnIndex)**: This method returns the name of the column by specifying the index of the column
4	**getColumnNames()**: This method returns the array of all the column names of the table.
5	**getCount()**: This method returns the total number of rows in the cursor

6	**getPosition()**: This method returns the current position of the cursor in the table
7	**isClosed()**: This method returns true if the cursor is closed and return false otherwise

Database - Helper class

For managing all the operations related to the database , an helper class has been given and is called SQLiteOpenHelper. It automatically manages the creation and update of the database. Its syntax is given below

```
public class DBHelper extends SQLiteOpenHelper {
   public DBHelper(){
      super(context,DATABASE_NAME,null,1);
   }
   public void onCreate(SQLiteDatabase db) {}
   public void onUpgrade(SQLiteDatabase database, int oldVersion, int newVersion) {}
}
```

Android - Emulator

The Android SDK includes a virtual mobile device emulator that runs on your computer. The emulator lets you prototype, develop and test Android applications without using a physical device.

Here we are going to explore different functionalities in the emulator that are present in the real android device.

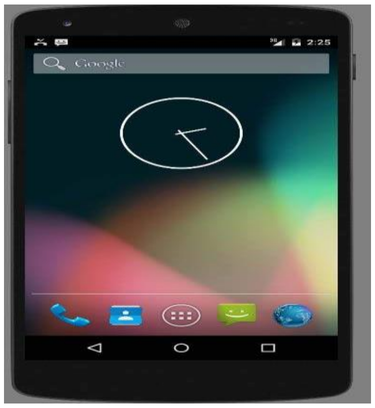
Creating AVD

If you want to emulate a real device, first crate an AVD with the same device configurations as real device, then launch this AVD from AVD manager.

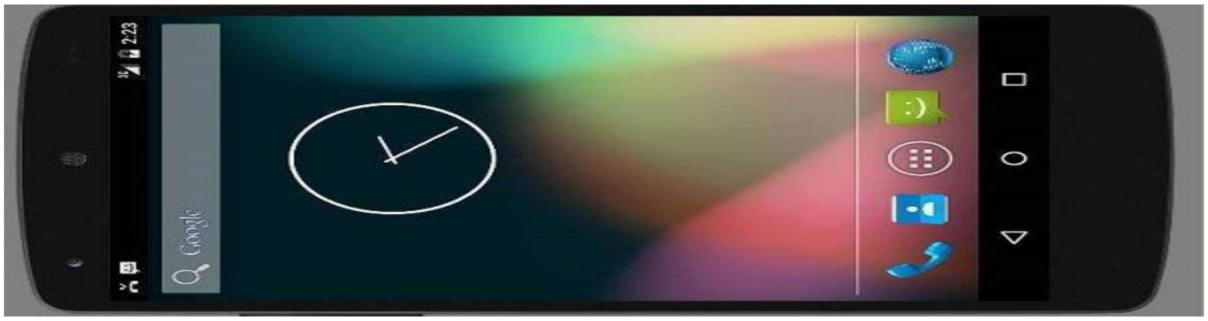
Changing Orientation

Usually by default when you launch the emulator, its orientation is vertical, but you can change it orientation by pressing Ctrl+F11 key from keyboard.

First launch the emulator. It is shown in the picture below −



Once it is launched, press **Ctrl+F11** key to change its orientation. It is shown below –



Emulator Commands

Apart from just orientation commands, there are other very useful commands of emulator that you should keep in mind while using emulator. They are listed below –

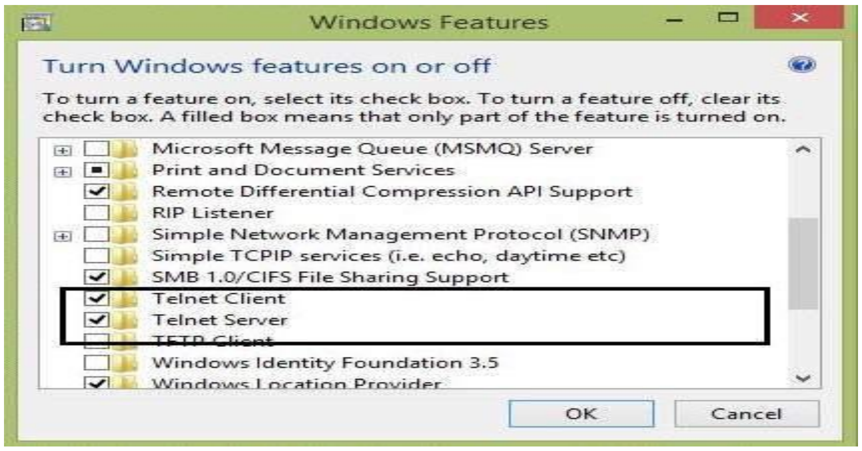
Sr.No	Command & description
1	**Home**: Shifts to main screen
2	**F2**: Toggles context sensitive menu
3	**F3**: Bring out call log
4	**F4**: End call
5	**F5**: Search

6	**F6**: Toggle trackball mode
7	**F7**: Power button
8	**F8**: Toggle data network
9	**Ctrl+F5**: Ring Volume up
10	**Ctrl+F6**: Ring Volume down

Emulator - Sending SMS

You can emulate sending SMS to your emulator. There are two ways to do that. You can do that from DDMS which can be found in Android studio, or from Telnet.(Network utility found in windows).

Sending SMS through Telnet


Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

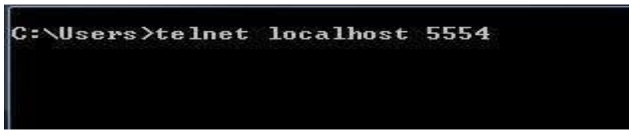
- ☐ Microsoft Message Queue (MSMQ) Server
- ■ Print and Document Services
- ✓ Remote Differential Compression API Support
- ☐ RIP Listener
- ☐ Simple Network Management Protocol (SNMP)
- ☐ Simple TCPIP services (i.e. echo, daytime etc)
- ✓ SMB 1.0/CIFS File Sharing Support
- ✓ Telnet Client
- ✓ Telnet Server
- ☐ TFTP Client
- ☐ Windows Identity Foundation 3.5
- ✓ Windows Location Provider

Telnet is not enabled by default in windows. You have to enable it to use it. Once enabled you can go to command prompt and start telnet by typing telnet.

In order to send SMS , note down the AVD number which can be found on the title bar of the emulator. It could be like this 5554 e.t.c. Once noted , type this command in command prompt.

telnet localhost 5554

Press enter when you type the command. It is shown below in the figure.

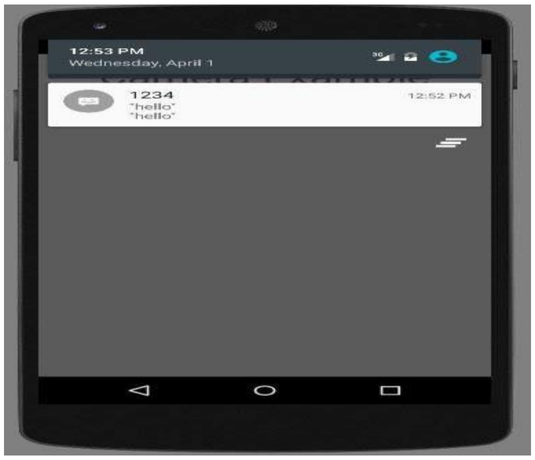


```
C:\Users>telnet localhost 5554
```

You will see that you are now connected to your emulator. Now type this command to send message.

sms send 1234 "hello"

Once you type this command , hit enter. Now look at the AVD. You will receive a notification displaying that you got a new text message. It is shown below –



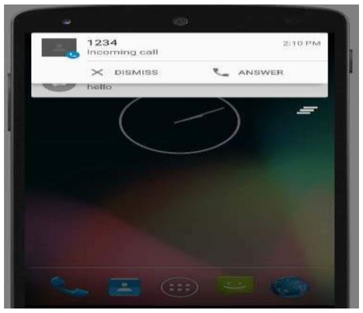
Emulator - Making Call

You can easily make phone calls to your emulator using telent client. You need to connect to your emulator from telnet. It is discussed in the sending sms topic above.

After that you will type this command in the telent window to make a call. Its syntax is given below –

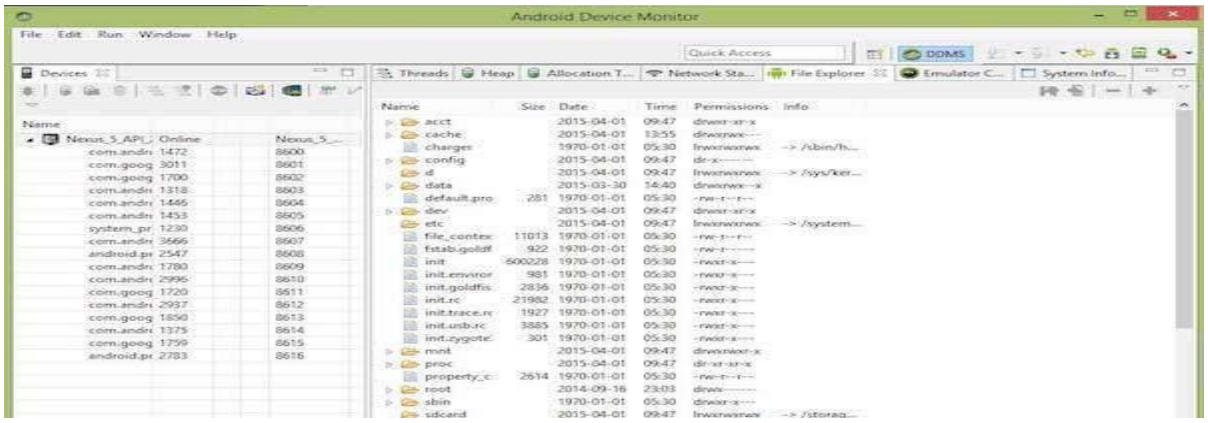
gsm call 1234

Once you type this command , hit enter. Now look at the AVD. You will receive a call from the number your put in the command. It is shown below –



Emulator - Transferring files

You can easily transfer files into the emulator and vice versa. In order to do that, you need to select the DDMS utility in Android studio. After that select the file explorer tab. It is shown below −



```
Name
Nexus_5_API_  Online   Nexus_5_...
  com.andr   1472      8600
  com.goog   3011      8601
  com.goog   1700      8602
  com.andr   1318      8603
  com.andr   1446      8604
  com.andr   1453      8605
  system_pr  1230      8606
  com.andr   3666      8607
  android.pr 2547      8608
  com.andr   1780      8609
  com.andr   2996      8610
  com.goog   1720      8611
  com.andr   2937      8612
  com.goog   1850      8613
  com.andr   1375      8614
  com.goog   1759      8615
  android.pr 2783      8616
```

```
Name              Size    Date        Time   Permissions  Info
acct                      2015-04-01  09:47  drwxr-xr-x
cache                     2015-04-01  13:55  drwxrwx---
charger                   1970-01-01  05:30  lrwxrwxrwx   -> /sbin/h...
config                    2015-04-01  09:47  dr-x------
d                         2015-04-01  09:47  lrwxrwxrwx   -> /sys/ker...
data                      2015-03-30  14:40  drwxrwx--x
default.pro        281    1970-01-01  05:30  -rw-r--r--
dev                       2015-04-01  09:47  drwxr-xr-x
etc                       2015-04-01  09:47  lrwxrwxrwx   -> /system...
file_contex      11013    1970-01-01  05:30  -rw-r--r--
fstab.goldf        922    1970-01-01  05:30  -rw-r-----
init            600228    1970-01-01  05:30  -rwxr-x---
init.enviror       981    1970-01-01  05:30  -rwxr-x---
init.goldfis      2836    1970-01-01  05:30  -rwxr-x---
init.rc          21982    1970-01-01  05:30  -rwxr-x---
init.trace.rc     1927    1970-01-01  05:30  -rwxr-x---
init.usb.rc       3885    1970-01-01  05:30  -rwxr-x---
init.zygote        301    1970-01-01  05:30  -rwxr-x---
mnt                       2015-04-01  09:47  drwxrwxr-x
proc                      2015-04-01  09:47  dr-xr-xr-x
property_c        2614    1970-01-01  05:30  -rw-r--r--
root                      2014-09-16  23:03  drwx------
sbin                      1970-01-01  05:30  drwxr-x---
sdcard                    2015-04-01  09:47  lrwxrwxrwx   -> /storag...
```

Browse through the explorer and make new folder , view existing contents e.t.c.

Android - Facebook Integration

Android allows your application to connect to facebook and share data or any kind of updates on facebook. There are two ways through which you can integrate facebook and share something from your application. These ways are listed below −

- Facebook SDK
- Intent Share

Integrating Facebook SDK

This is the first way of connecting with facebook. You have to register your application and then receive some Application Id , and then you have to download the facebook SDK and add it to your project. The steps are listed below:

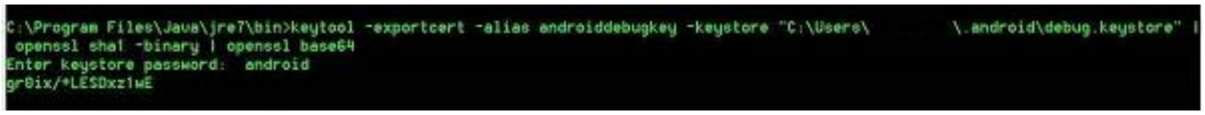
Generating application signature

You have to generate a key signature, but before you generate it, make sure you have SSL installed; otherwise you have to download SSl.

Now open command prompt and redirect to your java jre folder. Once you reach there, type this command exactly. You have to replace the path in the inverted commas with your keystore path which you can found in eclipse by selecting the window tab and selecting the preferences tab and then selecting the build option under android from left side.

```
keytool -exportcert -alias androiddebugkey -keystore "your path"
  | openssl sha1 -binary | openssl base64
```

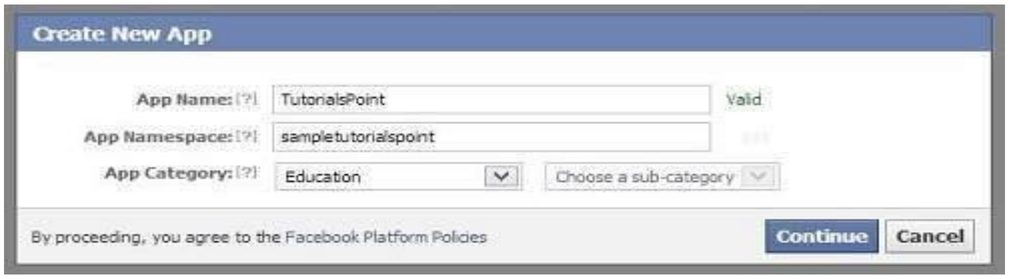
Once you enter it, you will be prompt for password. Give android as the password and then copy the key that is given to you. It is shown in the image below −



```
C:\Program Files\Java\jre7\bin>keytool -exportcert -alias androiddebugkey -keystore "C:\Users\        \.android\debug.keystore" |
 openssl sha1 -binary | openssl base64
Enter keystore password:  android
gr0ix/*LESDxz1wE
```

Registering your application

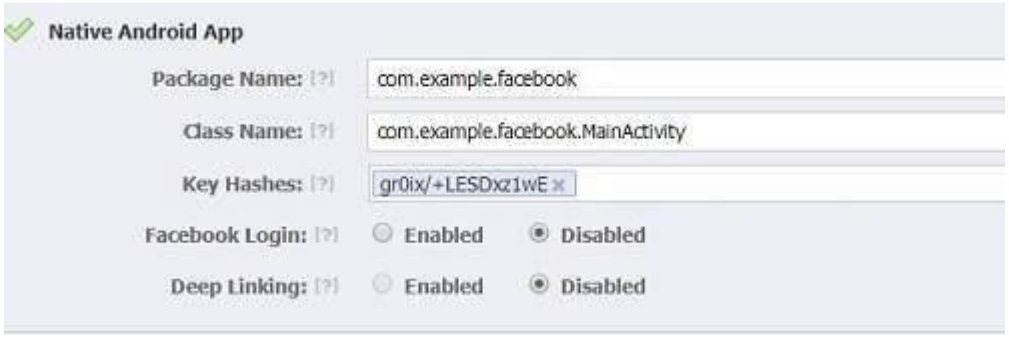
Now create a new facebook application at developers.facebook.com/apps and fill all the information. It is shown below −



App Name: [?] TutorialsPoint Valid
App Namespace: [?] sampletutorialspoint
App Category: [?] Education | Choose a sub-category

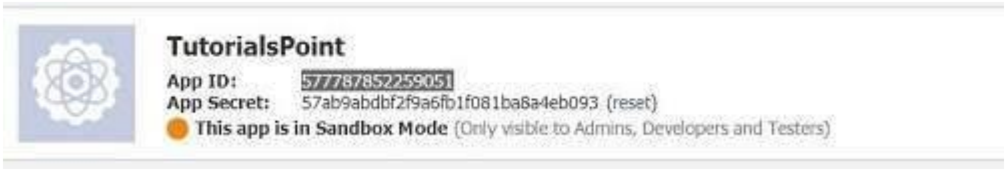
By proceeding, you agree to the Facebook Platform Policies

Now move to the native android app section and fill in your project and class name and paste the hash that you copied in step 1. It is shown below −



Package Name: [?] com.example.facebook
Class Name: [?] com.example.facebook.MainActivity
Key Hashes: [?] gr0ix/+LESDxz1wE
Facebook Login: [?] Enabled [✓ Disabled]
Deep Linking: [?] Enabled [✓ Disabled]

If everything works fine, you will receive an application ID with the secret. Just copy the application id and save it somewhere. It is shown in the image below −



TutorialsPoint
App ID: 577787852259051
App Secret: 57ab9abdbf2f9a6fb1f081ba8a4eb093 (reset)
This app is in Sandbox Mode (Only visible to Admins, Developers and Testers)

Downloading SDK and integrating it

Download facebook sdk here. Import this into eclipse. Once imported, right click on your facebook project and click on properties.Click on android, click on add button and select facebook sdk as the project.Click ok.

Creating facebook login application

Once everything is complete , you can run the samples, that comes with SDK or create your own application. In order to login, you need to call **openActiveSession** method and implements its callback. Its syntax is given below −

```
// start Facebook Login
Session.openActiveSession(this, true, new Session.StatusCallback() {
  // callback when session changes state
  public void call(Session session, SessionState state, Exception exception) {
    if (session.isOpened()) {
      // make request to;2 the /me API
      Request.executeMeRequestAsync(session, new Request.GraphUserCallback() {
        // callback after Graph API response with user object
        @Override
        public void onCompleted(GraphUser user, Response response) {
          if (user != null) {
            TextView welcome = (TextView) findViewById(R.id.welcome);
            welcome.setText("Hello " + user.getName() + "!");
          }
        }
      });
    }
  }
}
```

Intent share

Intent share is used to share data between applications. In this strategy, we will not handle the SDK stuff, but let the facebook application handles it. We will simply call the facebook application and pass the data to share. This way, we can share something on facebook.

Android provides intent library to share data between activities and applications. In order to use it as share intent , we have to specify the type of the share intent to **ACTION_SEND**. Its syntax is given below −

```
Intent shareIntent = new Intent();
shareIntent.setAction(Intent.ACTION_SEND);
```

Next thing you need to is to define the type of data to pass , and then pass the data. Its syntax is given below −

```
shareIntent.setType("text/plain");
shareIntent.putExtra(Intent.EXTRA_TEXT, "Hello, from tutorialspoint");
startActivity(Intent.createChooser(shareIntent, "Share your thoughts"));
```

Apart from the these methods, there are other methods available that allows intent handling. They are listed below −

Sl.No	Method & description
1	**addCategory(String category)**: This method add a new category to the intent.
2	**createChooser(Intent target, CharSequence title)**: Convenience function for creating a ACTION_CHOOSER Intent
3	**getAction()**: This method retrieve the general action to be performed, such as ACTION_VIEW
4	**getCategories()**: This method return the set of all categories in the intent and the current scaling event
5	**putExtra(String name, int value)**: This method add extended data to the intent.
6	**toString()**: This method returns a string containing a concise, human-readable description of this object

Android - Google Maps

Android allows us to integrate google maps in our application. You can show any location on the map , or can show different routes on the map e.t.c. You can also customize the map according to your choices.

Google Map - Layout file

Now you have to add the map fragment into xml layout file. Its syntax is given below −

```
<fragment
  android:id="@+id/map"
  android:name="com.google.android.gms.maps.MapFragment"
  android:layout_width="match_parent"
  android:layout_height="match_parent"/>
```

Google Map - AndroidManifest file

The next thing you need to do is to add some permissions along with the Google Map API key in the AndroidManifest.XML file. Its syntax is given below −

```
<!--Permissions-->

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="com.google.android.providers.gsf.permission.
  READ_GSERVICES" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<!--Google MAP API key-->

<meta-data
  android:name="com.google.android.maps.v2.API_KEY"
  android:value="AIzaSyDKymeBXNeiFWY5jRUejv6zItpmr2MVyQ0" />
```

Customizing Google Map

You can easily customize google map from its default view , and change it according to your demand.

Adding Marker

You can place a maker with some text over it displaying your location on the map. It can be done by via **addMarker()** method. Its syntax is given below −

```
final LatLng TutorialsPoint = new LatLng(21 , 57);
Marker TP = googleMap.addMarker(new MarkerOptions()
  .position(TutorialsPoint).title("TutorialsPoint"));
```

Changing Map Type

You can also change the type of the MAP. There are four different types of map and each give a different view of the map. These types are Normal,Hybrid,Satellite and terrain. You can use them as below

```
googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
googleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
googleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
```

Enable/Disable zoom

You can also enable or disable the zoom gestures in the map by calling the **setZoomControlsEnabled(boolean)** method. Its syntax is given below −

```
googleMap.getUiSettings().setZoomGesturesEnabled(true);
```

Apart from these customization, there are other methods available in the GoogleMap class , that helps you more customize the map. They are listed below −

Sl.No	Method & description
1	**addCircle(CircleOptions options)**: This method add a circle to the map
2	**addPolygon(PolygonOptions options)**: This method add a polygon to the map
3	**addTileOverlay(TileOverlayOptions options)**: This method add tile overlay to the map
4	**animateCamera(CameraUpdate update)**: This method Moves the map according to the update with an animation
5	**clear()**: This method removes everything from the map.
6	**getMyLocation()**: This method returns the currently displayed user location.
7	**moveCamera(CameraUpdate update)**: This method repositions the camera according to the instructions defined in the update
8	**setTrafficEnabled(boolean enabled)**: This method Toggles the traffic layer on or off.
9	**snapshot(GoogleMap.SnapshotReadyCallback callback)**: This method Takes a snapshot of the map
10	**stopAnimation()**: This method stops the camera animation if there is one in progress

Android - Image Effects

Android allows you to manipulate images by adding different kinds of effects on the images. You can easily apply image processing techniques to add certain kinds of effects on images. The effects could be brightness,darkness, grayscale conversion e.t.c.

Android provides Bitmap class to handle images. This can be found under android.graphics.bitmap. There are many ways through which you can instantiate bitmap. We are creating a bitmap of image from the imageView.

```
private Bitmap bmp;
private ImageView img;
img = (ImageView)findViewById(R.id.imageView1);
BitmapDrawable  abmp = (BitmapDrawable)img.getDrawable();
```

Now we will create bitmap by calling getBitmap() function of BitmapDrawable class. Its syntax is given below −

```
bmp = abmp.getBitmap();
```

An image is nothing but a two dimensional matrix. Same way you will handle a bitmap. An image consist of pixels. So you will get pixels from this bitmap and apply processing to it. Its syntax is as follows −

```
for(int i=0; i<bmp.getWidth(); i++){
  for(int j=0; j<bmp.getHeight(); j++){
    int p = bmp.getPixel(i, j);
  }
}
```

The getWidth() and getHeight() functions returns the height and width of the matrix. The getPixel() method returns the pixel at the specified index. Once you got the pixel, you can easily manipulate it according to your needs.

Apart from these methods, there are other methods that help us manipulate images better.

Sl.No	Method & description
1	**copy(Bitmap.Config config, boolean isMutable)**: This method copy this bitmap's pixels into the new bitmap
2	**createBitmap(DisplayMetrics display, int width, int height, Bitmap.Config config)**: Returns a mutable bitmap with the specified width and height

3	**createBitmap(int width, int height, Bitmap.Config config)**: Returns a mutable bitmap with the specified width and height
4	**createBitmap(Bitmap src)**: Returns an immutable bitmap from the source bitmap
5	**extractAlpha()**: Returns a new bitmap that captures the alpha values of the original
6	**getConfig()**: This mehtod eturn that config, otherwise return null
7	**getDensity()**: Returns the density for this bitmap
8	**getRowBytes()**: Return the number of bytes between rows in the bitmap's pixels
9	**setPixel(int x, int y, int color)**: Write the specified Color into the bitmap (assuming it is mutable) at the x,y coordinate
10	**setDensity(int density)**: This method specifies the density for this bitmap

Android - Internal Storage

Android provides many kinds of storage for applications to store their data. These storage places are shared preferences, internal and external storage, SQLite storage, and storage via network connection. Internal storage is the storage of the private data on the device memory.

By default these files are private and are accessed by only your application and get deleted , when user delete your application.

Writing file: In order to use internal storage to write some data in the file, call the openFileOutput() method with the name of the file and the mode. The mode could be private , public e.t.c. Its syntax is given below −

```
FileOutputStream fOut = openFileOutput("file name here",MODE_WORLD_READABLE);
```

The method openFileOutput() returns an instance of FileOutputStream. So you receive it in the object of FileInputStream. After that you can call write method to write data on the file. Its syntax is given below −

```
String str = "data";
fOut.write(str.getBytes());
fOut.close();
```

Reading file: In order to read from the file you just created , call the openFileInput() method with the name of the file. It returns an instance of FileInputStream. Its syntax is given below −

```
FileInputStream fin = openFileInput(file);
```

After that, you can call read method to read one character at a time from the file and then you can print it. Its syntax is given below −

```
int c;
String temp="";
while( (c = fin.read()) != -1){
  temp = temp + Character.toString((char)c);
}
//string temp contains all the data of the file.
fin.close();
```

Apart from the the methods of write and close, there are other methods provided by the **FileOutputStream** class for better writing files. These methods are listed below −

Sl.No	Method & description
1	**FileOutputStream(File file, boolean append)**: This method constructs a new FileOutputStream that writes to file.
2	**getChannel()**: This method returns a write-only FileChannel that shares its position with this stream
3	**getFD()**: This method returns the underlying file descriptor
4	**write(byte[] buffer, int byteOffset, int byteCount)**: This method Writes count bytes from the byte array buffer starting at position offset to this stream

Apart from the the methods of read and close, there are other methods provided by the **FileInputStream** class for better reading files. These methods are listed below –

Sl.No	Method & description
1	**available()**: This method returns an estimated number of bytes that can be read or skipped without blocking for more input
2	**getChannel()**: This method returns a read-only FileChannel that shares its position with this stream
3	**getFD()**: This method returns the underlying file descriptor
4	**read(byte[] buffer, int byteOffset, int byteCount)**: This method reads at most length bytes from this stream and stores them in the byte array b starting at offset

Android - Login Screen

A login application is the screen asking your credentials to login to some particular application. You might have seen it when logging into facebook, twitter etc.,

First you have to define two TextView asking username and password of the user. The password TextView must have **inputType** set to password. Its syntax is given below −

```
<EditText
  android:id = "@+id/editText2"
  android:layout_width = "wrap_content"
  android:layout_height = "wrap_content"
  android:inputType = "textPassword" />

<EditText
  android:id = "@+id/editText1"
  android:layout_width = "wrap_content"
  android:layout_height = "wrap_content"
/>
```

Define a button with login text and set its **onClick** Property. After that define the function mentioned in the onClick property in the java file.

```
<Button
  android:id = "@+id/button1"
  android:layout_width = "wrap_content"
  android:layout_height = "wrap_content"
  android:onClick = "login"
  android:text = "@string/Login"
/>
```

In the java file, inside the method of onClick get the username and passwords text using **getText()** and **toString()** method and match it with the text using **equals()** function.

```
EditText username = (EditText)findViewById(R.id.editText1);
EditText password = (EditText)findViewById(R.id.editText2);

public void login(View view){
  if(username.getText().toString().equals("admin") &&
password.getText().toString().equals("admin")){
  //correcct password
  }else{
  //wrong password
}
```

The last thing you need to do is to provide a security mechanism, so that unwanted attempts should be avoided. For this initialize a variable and on each false attempt, decrement it. And when it reaches to 0, disable the login button.

```
int counter = 3;
counter--;

if(counter==0){
  //disble the button, close the application e.t.c
}
```

Android - MediaPlayer

Android provides many ways to control playback of audio/video files and streams. One of this way is through a class called **MediaPlayer**.

Android is providing MediaPlayer class to access built-in mediaplayer services like playing audio,video e.t.c. In order to use MediaPlayer, we have to call a static Method **create()** of this class. This method returns an instance of MediaPlayer class. Its syntax is as follows −

MediaPlayer mediaPlayer = MediaPlayer.create(this, R.raw.song);

The second parameter is the name of the song that you want to play. You have to make a new folder under your project with name **raw** and place the music file into it.

Once you have created the Mediaplayer object you can call some methods to start or stop the music. These methods are listed below.

```
mediaPlayer.start();
mediaPlayer.pause();
```

On call to **start()** method, the music will start playing from the beginning. If this method is called again after the **pause()** method, the music would start playing from where it is left and not from the beginning.

In order to start music from the beginning, you have to call **reset()** method. Its syntax is given below.

mediaPlayer.reset();

Apart from the start and pause method, there are other methods provided by this class for better dealing with audio/video files. These methods are listed below −

Sl.No	Method & description
1	**isPlaying()**: This method just returns true/false indicating the song is playing or not
2	**seekTo(position)**: This method takes an integer, and move song to that particular position millisecond
3	**getCurrentPosition()**: This method returns the current position of song in milliseconds
4	**getDuration()**: This method returns the total time duration of song in milliseconds
5	**reset()**: This method resets the media player
6	**release()**: This method releases any resource attached with MediaPlayer object
7	**setVolume(float leftVolume, float rightVolume)**: This method sets the up down volume for this player
8	**setDataSource(FileDescriptor fd)**: This method sets the data source of audio/video file

9	**selectTrack(int index)**: This method takes an integer, and select the track from the list on that particular index
10	**getTrackInfo()**: This method returns an array of track information

Android – Multi touch

Multi-touch gesture happens when more than one finger touches the screen at the same time. Android allows us to detect these gestures.

Android system generates the following touch events whenever multiple fingers touches the screen at the same time.

Sl.No	Event & description
1	**ACTION_DOWN**: For the first pointer that touches the screen. This starts the gesture.
2	**ACTION_POINTER_DOWN**: For extra pointers that enter the screen beyond the first.
3	**ACTION_MOVE**: A change has happened during a press gesture.
4	**ACTION_POINTER_UP**: Sent when a non-primary pointer goes up.
5	**ACTION_UP**: Sent when the last pointer leaves the screen.

So in order to detect any of the above mention event , you need to override **onTouchEvent()** method and check these events manually. Its syntax is given below −

```
public boolean onTouchEvent(MotionEvent ev){
   final int actionPeformed = ev.getAction();

   switch(actionPeformed){
      case MotionEvent.ACTION_DOWN:{
         break;
      }

      case MotionEvent.ACTION_MOVE:{
         break;
      }
      return true;
   }
}
```

In these cases, you can perform any calculation you like. For example zooming , shrinking e.t.c. In order to get the co-ordinates of the X and Y axis, you can call **getX()** and **getY()** method. Its syntax is given below −

```
final float x = ev.getX();
final float y = ev.getY();
```

Apart from these methods, there are other methods provided by this MotionEvent class for better dealing with multitouch. These methods are listed below −

Sr.No	Method & description
1	**getAction()**: This method returns the kind of action being performed
2	**getPressure()**: This method returns the current pressure of this event for the first index
3	**getRawX()**: This method returns the original raw X coordinate of this event
4	**getRawY()**: This method returns the original raw Y coordinate of this event
5	**getSize()**: This method returns the size for the first pointer index
6	**getSource()**: This method gets the source of the event
7	**getXPrecision()**: This method return the precision of the X coordinates being reported

8	**getYPrecision()**: This method return the precision of the Y coordinates being reported

Android - Navigation

Here we will discuss, how we can provide navigation forward and backward between an application. We will first look at how to provide up navigation in an application.

Providing Up Navigation

The up navigation will allow our application to move to previous activity from the next activity. It can be done like this.

To implement Up navigation, the first step is to declare which activity is the appropriate parent for each activity. You can do it by specifying **parentActivityName** attribute in an activity. Its syntax is given below −

android:parentActivityName = "com.example.test.MainActivity"

After that you need to call **setDisplayHomeAsUpEnabled** method of **getActionBar()** in the onCreate method of the activity. This will enable the back button in the top action bar.

```
getActionBar().setDisplayHomeAsUpEnabled(true);
```

The last thing you need to do is to override **onOptionsItemSelected** method. when the user presses it, your activity receives a call to onOptionsItemSelected(). The ID for the action is **android.R.id.home**.Its syntax is given below −

```
public boolean onOptionsItemSelected(MenuItem item) {

  switch (item.getItemId()) {
    case android.R.id.home:
    NavUtils.navigateUpFromSameTask(this);
    return true;
  }
}
```

Handling device back button

Since you have enabled your back button to navigate within your application, you might want to put the application close function in the device back button.

It can be done by overriding **onBackPressed** and then calling **moveTaskToBack** and **finish** method. Its syntax is given below −

```
@Override
```

```
public void onBackPressed() {
   moveTaskToBack(true);
   MainActivity2.this.finish();
}
```

Apart from this setDisplayHomeAsUpEnabled method, there are other methods available in ActionBar API class. They are listed below −

Sl.No	Method & description
1	**addTab(ActionBar.Tab tab, boolean setSelected)**: This method adds a tab for use in tabbed navigation mode
2	**getSelectedTab()**: This method returns the currently selected tab if in tabbed navigation mode and there is at least one tab present
3	**hide()**: This method hide the ActionBar if it is currently showing
4	**removeAllTabs()**: This method remove all tabs from the action bar and deselect the current tab
5	**selectTab(ActionBar.Tab tab)**: This method select the specified tab

Android - PHP/MYSQL

This is very useful in case we have a webserver, and you want to access its data on your android application. MYSQL is used as a database at the webserver and PHP is used to fetch data from the database. Our application will communicate with the PHP page with necessary parameters and PHP will contact MYSQL database and will fetch the result and return the results to us.

PHP - MYSQL

Creating Database

MYSQL database can be created easily using this simple script. The **CREATE DATABASE** statement creates the database.

```
<?php
   $con=mysqli_connect("example.com","username","password");
```

```
  $sql="CREATE DATABASE my_db";
  if (mysqli_query($con,$sql)) {
    echo "Database my_db created successfully";
  }
?>
```

Creating Tables

Once database is created, its time to create some tables in the database. The **CREATE TABLE** statement creates the database.

```
<?php
  $con=mysqli_connect("example.com","username","password","my_db");
  $sql="CREATE TABLE table1(Username CHAR(30),Password CHAR(30),Role
CHAR(30))";
  if (mysqli_query($con,$sql)) {
    echo "Table have been created successfully";
  }
?>
```

Inserting Values in tables

When the database and tables are created. Now its time to insert some data into the tables. The **Insert Into** statement creates the database.

```
<?php
  $con=mysqli_connect("example.com","username","password","my_db");
  $sql="INSERT INTO table1 (FirstName, LastName, Age) VALUES ('admin',
'admin','adminstrator')";
  if (mysqli_query($con,$sql)) {
    echo "Values have been inserted successfully";
  }
?>
```

PHP - GET and POST methods

PHP is also used to fetch the record from the mysql database once it is created. In order to fetch record some information must be passed to PHP page regarding what record to be fetched.

The first method to pass information is through GET method in which **$_GET** command is used. The variables are passed in the url and the record is fetched. Its syntax is given below −

```
<?php
  $con=mysqli_connect("example.com","username","password","database name");

  if (mysqli_connect_errno($con)) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }
  $username = $_GET['username'];
  $password = $_GET['password'];
  $result = mysqli_query($con,"SELECT Role FROM table1 where Username='$username'
    and Password='$password'");
  $row = mysqli_fetch_array($result);
  $data = $row[0];

  if($data){
    echo $data;
  }
  mysqli_close($con);
?>
```

The second method is to use POST method. The only change in the above script is to replace $_GET with **$_POST**. In Post method, the variables are not passed through URL.

Android - Connecting MYSQL

Connecting Via Get Method

There are two ways to connect to MYSQL via PHP page. The first one is called **Get method**. We will use **HttpGet** and **HttpClient** class to connect. Their syntax is given below −

```
URL url = new URL(link);
HttpClient client = new DefaultHttpClient();
HttpGet request = new HttpGet();
request.setURI(new URI(link));
```

After that you need to call **execute** method of HttpClient class and receive it in a HttpResponse object. After that you need to open streams to receive the data.

```
HttpResponse response = client.execute(request);
BufferedReader in = new BufferedReader
(new InputStreamReader(response.getEntity().getContent()));
```

Connecting Via Post Method

In the Post method, the **URLEncoder**,**URLConnection** class will be used. The urlencoder will encode the information of the passing variables. It's syntax is given below −

```
URL url = new URL(link);
String data  = URLEncoder.encode("username", "UTF-8")
+ "=" + URLEncoder.encode(username, "UTF-8");
data += "&" + URLEncoder.encode("password", "UTF-8")
```

```
+ "=" + URLEncoder.encode(password, "UTF-8");
URLConnection conn = url.openConnection();
```

The last thing you need to do is to write this data to the link. After writing, you need to open stream to receive the responded data.

```
OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());
wr.write( data );
BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
```

Android - Push Notification

A notification is a message you can display to the user outside of your application's normal UI. You can create your own notifications in android very easily.

Android provides **NotificationManager** class for this purpose. In order to use this class, you need to instantiate an object of this class by requesting the android system through **getSystemService() method**. Its syntax is given below −

NotificationManager NM;
NM=(NotificationManager)getSystemService(Context.NOTIFICATION_SERVICE);

After that you will create Notification through **Notification** class and specify its attributes such as icon,title and time e.t.c. Its syntax is given below −

Notification notify = new
Notification(android.R.drawable.stat_notify_more,title,System.currentTimeMillis());

The next thing you need to do is to create a **PendingIntent** by passing context and intent as a parameter. By giving a PendingIntent to another application, you are granting it the right to perform the operation you have specified as if the other application was yourself.

PendingIntent pending = PendingIntent.getActivity(getApplicationContext(), 0, new Intent(),0);

The last thing you need to do is to call **setLatestEventInfo** method of the Notification class and pass the pending intent along with notification subject and body details. Its syntax is given below. And then finally call the notify method of the NotificationManager class.

notify.setLatestEventInfo(getApplicationContext(), subject, body,pending);
NM.notify(0, notify);

Apart from the notify method, there are other methods available in the NotificationManager class. They are listed below –

Sl.No	Method & description
1	**cancel(int id)**: This method cancel a previously shown notification.
2	**cancel(String tag, int id)**: This method also cancel a previously shown notification.
3	**cancelAll()**: This method cancel all previously shown notifications.
4	**notify(int id, Notification notification)**: This method post a notification to be shown in the status bar.
5	**notify(String tag, int id, Notification notification)**: This method also Post a notification to be shown in the status bar.

Android - SDK Manager

To download and install latest android APIs and development tools from the internet, android provide us with android SDK manager. Android SDK Manager separates the APIs, tools and different platforms into different packages which you can download.

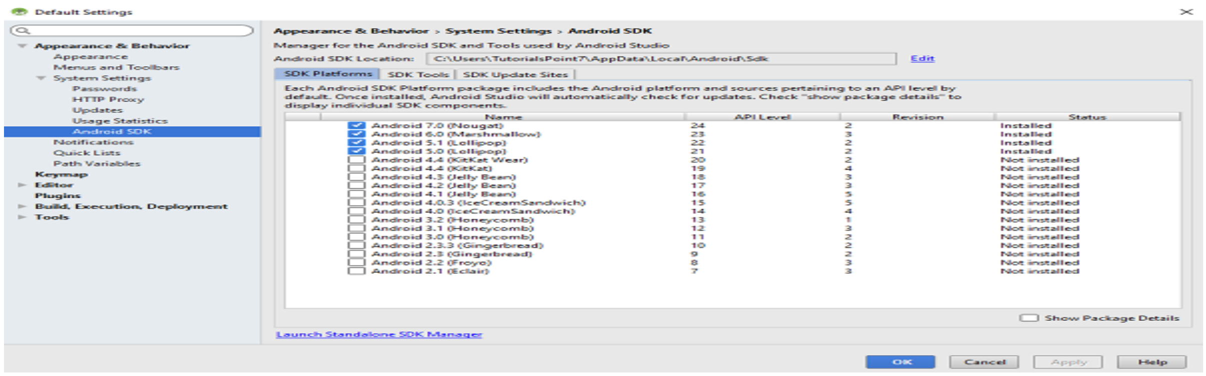
Android SDK manager comes with the Android SDK bundle. You can't download it separately.

Running Android SDK Manager

Once downloaded, you can launch Android SDK Manager in one of the following ways −

- Click **tools->Android-> SDK Manager** option in Eclipse.
- Double Click on the **SDK Manager.exe** file in the Android SDK folder.

When it runs you will see the following screen −



Appearance & Behavior > System Settings > Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\TutorialsPoint7\AppData\Local\Android\Sdk

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
✓	Android 7.0 (Nougat)	24	2	Installed
✓	Android 6.0 (Marshmallow)	23	3	Installed
✓	Android 5.1 (Lollipop)	22	2	Installed
✓	Android 5.0 (Lollipop)	21	2	Installed
☐	Android 4.4 (KitKat Wear)	20	2	Not installed
☐	Android 4.4 (KitKat)	19	4	Not installed
☐	Android 4.3 (Jelly Bean)	18	3	Not installed
☐	Android 4.2 (Jelly Bean)	17	3	Not installed
☐	Android 4.1 (Jelly Bean)	16	5	Not installed
☐	Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
☐	Android 4.0 (IceCreamSandwich)	14	4	Not installed
☐	Android 3.2 (Honeycomb)	13	1	Not installed
☐	Android 3.1 (Honeycomb)	12	3	Not installed
☐	Android 3.0 (Honeycomb)	11	2	Not installed
☐	Android 2.3.3 (Gingerbread)	10	2	Not installed
☐	Android 2.3 (Gingerbread)	9	2	Not installed
☐	Android 2.2 (Froyo)	8	3	Not installed
☐	Android 2.1 (Eclair)	7	3	Not installed

☐ Show Package Details

You can select which package you want to download by selecting the checkbox and then click **Install** to install those packages. By default SDK Manager keeps it up to date with latest APIs and other packages.

Once you download the SDK, following packages are available but first three are necessary to run your SDK and others are recommended.

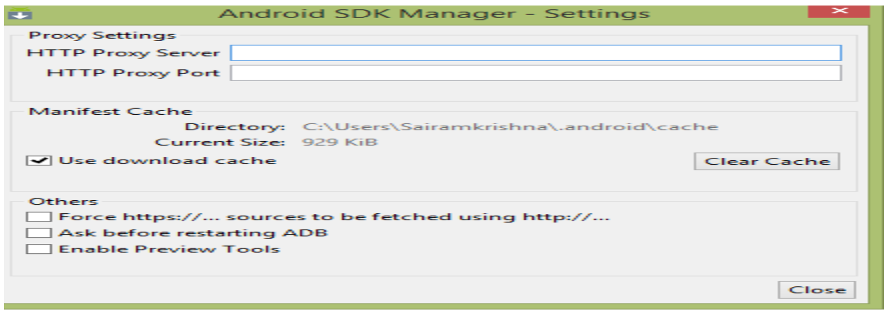
Recommended Packages

Sr.No	Package & Description
1	**SDK Tools**: This is necessary package to run your SDK.
2	**SDK Platform-tools**: This package will be installed once when you first run the SDK manager.
3	**SDK Platform**: At least one platform must be installed in your environment to run your application.
4	**System Image**: It's a good practice to download system images for all of the android versions so you can test your app on them with the Android Emulator.
5	**SDK Samples**: This will give you some sample codes to learn about android.

Enabling Proxy in Android SDK Manager

When you run the Android SDK Manager, by default it will check from the Android Repository and Third Party Add-ons and display the available packages to you.

If you want to use proxy, you can do it by clicking on the **Tools-->Options**in the menu. Once you click it, you will see the following screen –



Directory: C:\Users\Sairamkrishna\.android\cache
Current Size: 929 KiB
☑ Use download cache
☐ Force https://... sources to be fetched using http://...
☐ Ask before restarting ADB
☐ Enable Preview Tools

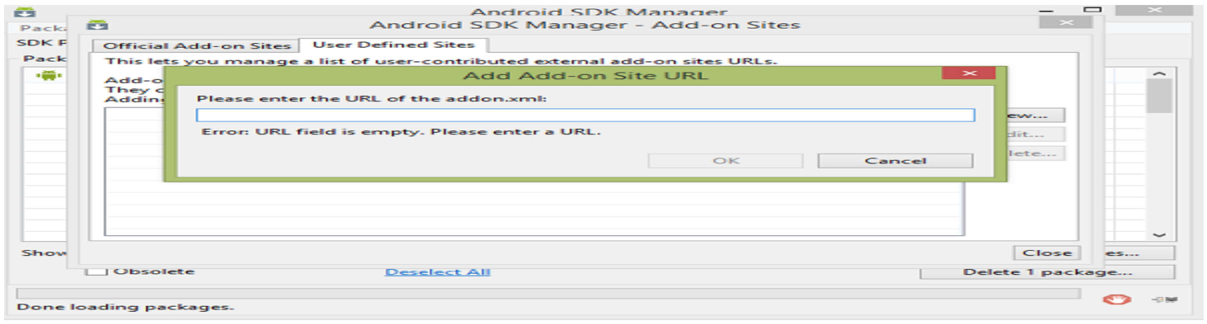
Just Enter the proxy and run your SDK Manager.

Adding New Third Party Sites

If you want to download some Third Party made Android add-ons, you can do it in the SDK manager by following steps −

- Click on the **Tools** option in the menu.
- Click on the **Manage Add-On Sites** option in the sub menu.
- Select the **User Defined Sites** tab.
- Click the **New** button.

Following screen will be displayed −



This lets you manage a list of user-contributed external add-on sites URLs.

Please enter the URL of the addon.xml:

Error: URL field is empty. Please enter a URL.

Done loading packages.

Just add the URL of Add-on.xml file and click **Ok**. Now you can download the Third Party Add-on in your development environment and use it.

Android - Session Management

Session help you when want to store user data outside your application, so that when the next time user use your application, you can easily get back his details and perform accordingly.

This can be done in many ways. But the easiest and nicest way of doing this is through **Shared Preferences**.

Shared Preferences

Shared Preferences allow you to save and retrieve data in the form of key,value pair. In order to use shared preferences, you have to call a method getSharedPreferences() that returns a SharedPreference instance pointing to the file that contains the values of preferences.

SharedPreferences sharedpreferences = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE);

You can save something in the sharedpreferences by using SharedPreferences.Editor class. You will call the edit method of SharedPreference instance and will receive it in an editor object. Its syntax is −

Editor editor = sharedpreferences.edit();
editor.putString("key", "value");

editor.commit();

Apart from the putString method, there are methods available in the editor class that allows manipulation of data inside shared preferences. They are listed as follows −

Sl.No	Mode & description
1	**apply()**: It is an abstract method. It will commit your changes back from editor to the sharedPreference object you are calling
2	**clear()**: It will remove all values from the editor
3	**remove(String key)**: It will remove the value whose key has been passed as a parameter
4	**putLong(String key, long value)**: It will save a long value in a preference editor
5	**putInt(String key, int value):** It will save a integer value in a preference editor
6	**putFloat(String key, float value)**: It will save a float value in a preference editor