

Extract text on image and translate android app source code is available now.

See example in Play Store



# Inducesmile

Mobile developer community for beginners and students

[ANDROID](#)
[KOTLIN](#)
[IOS \(SWIFT\)](#)
[HYBRID APP ▼](#)
[MOBILE GAMES](#)
[IOT - VR](#)
[TIPS](#)
[CODING HELP](#)
[BUY CODE](#)

[Home](#) ▶ [Android](#) ▶ JSON Feed Reader in Android using ListView



## JSON FEED READER IN ANDROID USING LISTVIEW

In this tutorial, we will take step by step look into building a JSON feed reader in Android. All the source code used in this example, are open source. You may grab a copy of the project source code from the download link provided. This app will fetch the recent posts from your php server eg wamp server and display list of posts along with post title, thumbnail, description on screen. You can also view details of each feed, share the article and can view the article on original website using Android WebView.

You will learn following things in this article.

1. Designing User Interface
2. Using AsyncTask to Download DataFrom Remote Server
3. JSON Parsing In Android
4. Download Image Asynchronously and Display it in Android ListView



Scan & Translate Android App  
Source code

[VIEW APP SAMPLE](#)

[WRITE FOR US](#)

## ANDROID SOURCE CODE POLL

Which of the below listed  
source codes should we publish  
next month?

Multi Restaurant Food  
Ordering App (27%, 39 Votes)

Stock and Inventory App  
(18%, 26 Votes)

Android Dating App (16%, 23

Accept

Before we start, the first thing I will do is to list the environment and tools I used in this android tutorial but feel free to use whatever environment or tools you are familiar with.

Windows 7

Android Studio

Samsung Galaxy Fame LiteMin

SDK 14 Target SDK 19

To create a new android application project, following the steps as stipulated below.

Go to File menu

Click on New menu

Click on Android Application

Enter Project name:Feed Reader

Package: com.inducesmile.feedreader

Keep other default selections.

Continue to click on next button until Finish button is active, then click on Finish Button. Once you are done with creating your project, make sure you change the package name if you did not use the same package.

## 1. DESIGNING USER INTERFACE

Here in this example, we are using three Activities. While first Activity is used to list out all feed items on a ListView, Second one is used to show the image preview and the description of each feed. Another Activity is used to view the original feed item on a WebView.

Below are the listed three activity classes

Android E-Book App (12%, 17 Votes)

☐

Appointment Booking App (10%, 15 Votes)

☐

Business & Service Provider Finder (6%, 9 Votes)

☐

Total Voters: **147**

## LIKE US ON FACEBOOK



**Induce smile**

5,305 likes

Like Page

Be the first of your friends to like this

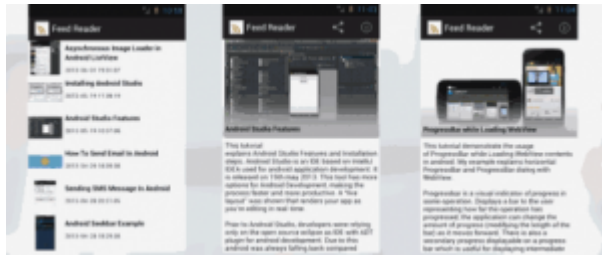
## RECENT POSTS

- Camera Translator All Language App
- Why MVVM and How to execute MVVM combined with Data Binding and LiveData in Four Simple Steps
- Camera Translator All Languages – Android Project idea source code for students
- Note Scan Android App -Project idea source code for students
- How To Build Android WhatsApp Clone Chat Application
- Android Car Rental App with Web Admin Panel – Project Idea Source Code For Students

Accept

\*com.inducesmile.feedreader.WebViewActivity

We will assume that you already have a little knowledge of creating basic user interfaces. We will be creating the layouts as shown in the images below.



- Android Retrofit 2 with JSON API Example
- Android Room with LiveData Example Tutorial
- Android UI App for Food Ordering and Delivery from Multiple Restaurants

On landing page of the app, It uses a ListView to display thumbnail image, title and date published, so here we need to create a ListView widget in our layout file. If you are not proficient on creating view layout elements, then I will suggest you visit my previous post on ***Android ListView Tutorial***.

For simplicity sake, our previous ListView Tutorial made use of a static data for the list. But in this example, I am fetching the data from php server and then displaying on a Listview. When fetching data from server, a loading ProgressBar will appear on screen which will let users know how far the download has gone and it will disappear once the list items are downloaded. Let's start with the code to build the user interface first.

## ACTIVITY\_POST\_LIST.XML

```
<?xml version="1.0" encoding="utf-8"?>
```

```
</>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
</>
```

```
<FrameLayout xmlns:android="http://schemas.android.com/a
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >
```

```
<ProgressBar
```

```
android:id="@+id/progressBar"
```

Accept

```

<ListView
    android:id="@ id/custom_list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:cacheColorHint="#00000000"
    android:dividerHeight="1dp"
    android:focusable="false"
    android:listSelector="@drawable/list_selector_flatcolor"
    android:visibility="gone" />

</FrameLayout>

```

In the above xml code, we are using FrameLayout. We will be displaying either of the UI widgets. During loading we will display ProgressBar and then we will display ListView after feed data is downloaded.

## LIST\_ROW\_LAYOUT.XML

This layout will be used for ListView row. Each Row is an RelativeLayout with an ImageView and two TextViews placed adjacent to the ImageView

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.co
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:minHeight="50dp"
    android:orientation="horizontal" >
    <ImageView
        android:id="@ id/thumbImage"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_alignParentLeft="true"
        android:layout_centerVertical="true"
        android:background="@drawable/list_placeholder" />

```

Accept

```

android:lineSpacingExtra="3dp"
android:paddingLeft="5dp"

android:paddingTop="5dp"
android:text=""
android:textColor="@drawable/list_item_text_selector"
android:textStyle="bold"
android:typeface="sans" />

<TextView
android:id="@ id/date"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@ id/title"
android:layout_toRightOf="@id/thumbImage"
android:paddingLeft="5dp"
android:paddingTop="5dp"
android:text=""
android:textColor="@drawable/list_item_text_selector"
android:textSize="11sp" />

</RelativeLayout>

```

In the above code, I have used "list\_item\_text\_selector" for changing the TextView color while list row is pressed.

## LIST\_ITEM\_TEXT\_SELECTOR.XML

```

<?xml version="1.0" encoding="UTF-8"?>

<selector xmlns:android="http://schemas.android.com/apk/
<item android:state_pressed="true"
android:color="@color/text_color_inverse" />
<item android:state_focused="true"
android:color="@color/text_color_inverse" />
<item android:color="@color/text_color_default" />
</selector>

```

We are ready with the layout for FeedList Screen. Now let's create a

Accept

```
public class CustomListAdapter extends BaseAdapter {</>
private ArrayList listData;
private LayoutInflater inflater;
private Context mContext;
public CustomListAdapter(Context context, ArrayList list
this.listData = listData;
inflater = LayoutInflater.from(context);
mContext = context;
}

@Override
public int getCount() {
return listData.size();
}

@Override
public Object getItem(int position) {
return listData.get(position);
}

@Override
public long getItemId(int position) {
return position;
}

public View getView(int position, View convertView, View
ViewHolder holder;
if (convertView == null) {
convertView = inflater.inflate(R.layout.list_row_1
holder = new ViewHolder();
holder.headlineView = (TextView) convertView.findViewByI
holder.reportedDateView = (TextView) convertView.findVie
holder.imageView = (ImageView) convertView.findViewById(
convertView.setTag(holder);
} else {
holder = (ViewHolder) convertView.getTag();
}

FeedItem newsItem = (FeedItem) listData.get(position);
holder.headlineView.setText(newsItem.getTitle());
holder.reportedDateView.setText(newsItem.getDate());
```

Accept

```
}

static class ViewHolder {
    TextView headlineView;
    TextView reportedDateView;
    ImageView imageView;
}
}
```

Now you must be getting some compilation error for FeedItem.

FeedItem is an model class used for reading feeds. It has the following fields with public getter and setter methods.

## FEEDITEM.JAVA

```
public class FeedItem implements Serializable {
    private String title;
    private String date;
    private String attachmentUrl;
    private String id;
    private String content;
    private String url;
}
```

Now, we are ready for the first screen layout. Let's move on to the FeedDetailsActivity layout. In this screen we have thumbnail image, title and the content to be displayed. Below is the layout code snippet.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffffff"
    android:orientation="vertical" >

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="200dp" >
```

Accept

```
</ImageView>

<TextView
    android:textStyle="bold" />
    android:id="@ id/title"
    android:layout_width="fill_parent"
    on
    android:layout_height="100dp"
    android:layout_alignParentBottom="true"
    android:background="@drawable/image_border"
    android:ellipsize="end"
    android:gravity="bottom"
    android:lineSpacingExtra="3dp"
    android:textColor="#00000c"
    android:maxLines="2"
    android:padding="5dp"
    android:text=""
</RelativeLayout>
<ScrollView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="5dp" >

    <TextView
        android:id="@ id/content"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textColor="#00000c" />
</ScrollView>
</LinearLayout>
```

In FeedDetailsActivity, we are using Android ActionBar commands.

## MENU.XML

```
<menu xmlns:android="http://schemas.android.com/apk/res/
    android:icon="@android:drawable/ic_menu_info_details"
    android:id="@ id/menu_view"
    <item
```

Accept



```

android:icon="@android:drawable/ic_menu_share"
on
android:id="@ id/menu_share"
<item
</menu>

```

Now we are done with all xml layouts. Let us have a look into the Activity components

## 2. ASYNCTASK TO DOWNLOAD DATA FROM SERVER

AsyncTask enables you to implement MultiThreading without getting your hands dirty with threads. AsyncTask enables proper and easy use of the UI thread. It allows performing background operations and passing the results to the UI thread. If we are executing a task that will take long to finish, for example downloading data over network, it is recommended to use AsyncTask.

### DOWNLOADFEEDTASK.JAVA

```

private class DownloadFilesTask extends AsyncTask<String,
@Override
protected void onProgressUpdate(Integer... values) {
}

@Override
protected void onPostExecute(Void result) {
if (null != feedList) {
updateList();
}
}

@Override
protected Void doInBackground(String... params) {
String url = params[0];

// getting JSON string from URL

```

Accept

```
}  
}
```

## DOWNLOADING JSON FROM SERVER

As downloading of feed data from Inducesmile is a long running task, we will execute it inside `doInBackground()` method. Once we have the data downloaded and parsed, then we can update the `ListView` with appropriate updated data.

```
public JSONObject getJSONFromUrl(String url) {  
    InputStream is = null;  
    JSONObject jsonObj = null;  
    String json = null;  
  
    // Making HTTP request  
    try {  
        // defaultHttpClient  
        DefaultHttpClient httpClient = new DefaultHttpClient();  
        HttpPost httpPost = new HttpPost(url);  
        HttpResponse httpResponse = httpClient.execute(httpPost)  
        HttpEntity httpEntity = httpResponse.getEntity();  
        is = httpEntity.getContent();  
  
        BufferedReader reader = new BufferedReader(new InputStre  
is, "iso-8859-1"), 8);  
        StringBuilder sb = new StringBuilder();  
        String line = null;  
        while ((line = reader.readLine()) != null) {  
            sb.append(line    "\n");  
        }  
        is.close();  
        json = sb.toString();  
    } catch (UnsupportedEncodingException e) {  
        e.printStackTrace();  
    } catch (ClientProtocolException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Accept

```

Log.e("JSON Parser", "Error parsing data " + e.toString()
}

// return JSON String
return jsonObj;
}

```

The above method will download the json feed and returned as a JSONObject. Now its the time to parse the JSON feed data I advice you to use localhost like wamp server for testing.

## JSON FEED PARSING IN ANDROID

The below Feed link gives the below JSON object structure.

```

{
  "status": "ok", "count": 10,  "category": { "id": 1, "tit
    "id": 1, "type": "post", "url": "https://inducesmile.
}

```

From this above structure we will be needing the title, date, url and attachment url. check out the code snippet below for json parsing.

```

public void parseJson(JSONObject json) {
try {

// parsing json object
if (json.getString("status").equalsIgnoreCase("ok")) {
JSONArray posts = json.getJSONArray("posts");

feedList = new ArrayList();

for (int i = 0; i < posts.length(); i ) {
JSONObject post = (JSONObject) posts.getJSONObject(i);
FeedItem item = new FeedItem();
item.setTitle(post.getString("title"));
item.setDate(post.getString("date"));
}
}
}
}

```

Accept

```
if (null != attachments && attachments.length() > 0) {  
    JSONObject attachment = attachments.getJSONObject(0);  
    if (attachment != null)  
        item.setAttachmentUrl(attachment.getString("url"));  
}  
  
feedList.add(item);  
}  
}  
} catch (JSONException e) {  
    e.printStackTrace();  
}  
}
```

## DOWNLOADING IMAGE ASYNCHRONOUSLY

A good practice to bring the best performance to android application is to make sure your main thread does the minimum amount of work.

Any long running tasks or heavy operations are usually performed in a different thread. Typical long running tasks could be network operations, reading files from memory, animations, etc.

Check out the below post for detailed implementation

[Android Remote Image Download using Android DownloadManager, Android AsyncTask, Volley, Universal Image Loader, Picasso and OkHttp](#)

## SHARE ARTICLE IN ANDROID

One of the best and most useful feature is sharing of information across different social networks. Android platform provides a handy way to share contents across different application using Share Intent. It lists out all of the available application that can handle the share event. Check out the code snippet

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

Finally, if you are looking for a way to load your website url on android webview rather than using a feed reader, we have written a tutorial that will guide you on how you can achieve it. This post will give you a complete understanding of loading external URL on android WebView with different configurations.

[How to load external web page inside Android WebView](#)

## USES OF FEED READER

- 1.Feed reader can be use to create website application, like this tutorial given is a web app feed reader.
- 2.feed reader can also be use to create tv app like mobdro, TV app tutorial we be drop soon.
- 3.Feed reader can also be use to create radio app it will also be drop soon.

You can download the code for this tutorial below. If you are having hard time downloading the tutorial, kindly contact me. Remember to subscribe with your email address to be among the first to receive my new android blog post once it is published.

## OTHER INTERESTING POSTS:

Android Studio – Add .Jar  
File as Android Library

Android GridView Vs  
GridLayout Example  
Tutorial

Accept

Google Page-level ads in  
Beta Integration

Android RecyclerView Swipe  
to dismiss and Item sorter  
using ItemTouchHelper and  
ItemTouchHelper.Simpl...

Create a beautiful Android  
Weather App using  
OpenWeatherMap API and  
Volley Library part 1

Android FirebaseUI  
Authentication Example


## ABOUT THE AUTHOR



**Adenuga558**

I Love Sharing Useful Article To Android Beginner



 **Inducesmile**  
**1,932**  
● 1 ● 9 ● 15

Accept

INDUCESMILE – ANDROID TUTORIAL, ANDROID APPS, ANDROID STUDIO, ANDROID SDK, ANDROID DEVELOPMENT COPYRIGHT © 2018.

[PROGRAMMING HELP](#)

[ANDROID TUTORIALS](#)

[CONTACT US](#)

[PRIVACY POLICY](#)

Accept