# Assignment 5

1. Google Books API in List View:

Activity:

**public class** IndexActivity **extends** AppCompatActivity  
 **implements** NavigationView.OnNavigationItemSelectedListener {  
  
 **private** Context **mContext**;  
 String **sharedText**;  
  
 RecyclerView **listView**;  
 **private** ListViewAdapter **bookAdapter**;  
 EditText **search**;  
  
 String **API\_URL** = **"https://www.googleapis.com/books/v1/volumes?q="**;  
  
 ArrayList<Book> **bookList** = **new** ArrayList<Book>();  
  
 **public** IndexActivity(){  
 **super**();  
 }  
  
 **public** IndexActivity(Context context){  
 **mContext** = context;  
 }  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_index***);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.***drawer\_layout***);  
 ActionBarDrawerToggle toggle = **new** ActionBarDrawerToggle(  
 **this**, drawer, toolbar, R.string.***navigation\_drawer\_open***, R.string.***navigation\_drawer\_close***);  
 drawer.setDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = (NavigationView) findViewById(R.id.***nav\_view***);  
 navigationView.setNavigationItemSelectedListener(**this**);  
  
 Intent intent = getIntent();  
 String action = intent.getAction();  
 String type = intent.getType();  
  
 **if** (Intent.***ACTION\_SEND***.equals(action) && type != **null**) {  
 **if** (**"text/plain"**.equals(type)) {  
 handleSendText(intent); *// Handle text being sent* }  
 }  
  
 }  
  
 **void** handleSendText(Intent intent) {  
 **sharedText** = intent.getStringExtra(Intent.***EXTRA\_TEXT***);  
 **if** (**sharedText** != **null** && !**""**.equalsIgnoreCase(**sharedText**)) {  
 NavigationView navigationView = (NavigationView) findViewById(R.id.***nav\_view***);  
 navigationView.setNavigationItemSelectedListener(**this**);  
 View header = navigationView.getHeaderView(0);  
  
 }  
 }  
  
 @Override  
 **public void** onBackPressed() {  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.***drawer\_layout***);  
 **if** (drawer.isDrawerOpen(GravityCompat.***START***)) {  
 drawer.closeDrawer(GravityCompat.***START***);  
 } **else** {  
 **super**.onBackPressed();  
 }  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***index***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.* **int** id = item.getItemId();  
  
 *//noinspection SimplifiableIfStatement* **if** (id == R.id.***action\_settings***) {  
 **return true**;  
 }  
  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings(**"StatementWithEmptyBody"**)  
 @Override  
 **public boolean** onNavigationItemSelected(MenuItem item) {  
 *// Handle navigation view item clicks here.* **int** id = item.getItemId();  
  
 **if** (id == R.id.***nav\_camera***) {  
 Intent sendIntent = **new** Intent(**this**,IndexActivity.**class**);  
 sendIntent.setAction(Intent.***ACTION\_SEND***);  
 sendIntent.putExtra(Intent.***EXTRA\_TEXT***, **sharedText**);  
 sendIntent.setType(**"text/plain"**);  
  
  
 **if**(sendIntent.resolveActivity(getPackageManager())!=**null**){  
 startActivity(sendIntent);  
 }  
 } **else if** (id == R.id.***nav\_gallery***) {  
 Intent sendIntent = **new** Intent(**this**,HelpActivity.**class**);  
 startActivity(sendIntent);  
  
  
 **if**(sendIntent.resolveActivity(getPackageManager())!=**null**){  
 startActivity(sendIntent);  
 }  
  
 } **else if** (id == R.id.***nav\_manage***) {  
 Intent sendIntent = **new** Intent(**this**,SettingsActivity.**class**);  
 sendIntent.setAction(Intent.***ACTION\_SEND***);  
 sendIntent.putExtra(Intent.***EXTRA\_TEXT***, **sharedText**);  
 sendIntent.setType(**"text/plain"**);  
  
  
 **if**(sendIntent.resolveActivity(getPackageManager())!=**null**){  
 startActivity(sendIntent);  
 }  
  
 } **else if** (id == R.id.***nav\_share***) {  
  
 }  
  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.***drawer\_layout***);  
 drawer.closeDrawer(GravityCompat.***START***);  
 **return true**;  
 }  
  
 **private class** BookSearchAsyncTaskRunner **extends** AsyncTask<String, String, String> {  
  
 **private** String **resp**;  
 ProgressBar **spinner**;  
  
 @Override  
 **protected** String doInBackground(String... strings) {  
  
 InputStream inputStream = **null**;  
 String result = **""**;  
 **try** {  
  
 *// create HttpClient* HttpClient httpclient = **new** DefaultHttpClient();  
  
 *// make GET request to the given URL* HttpResponse httpResponse = httpclient.execute(**new** HttpGet(strings[1]));  
  
 *// receive response as inputStream* inputStream = httpResponse.getEntity().getContent();  
  
 *// convert inputstream to string* **if**(inputStream != **null**) {  
 result = *convertInputStreamToString*(inputStream);  
 **try** {  
 **if**(result!=**null** && !**""**.equalsIgnoreCase(result)){  
  
 JSONObject object = **new** JSONObject(result);  
 JSONArray array = object.getJSONArray(**"items"**);  
 **for** (**int** i = 0; i <=5; i++) {  
  
 JSONObject item = array.getJSONObject(i);  
  
 String title = **""**;  
 String subtitle = **""**;  
 String author = **""**;  
 String imageLink = **""**;  
 String publishedDate=**""**;  
  
 **if**(item.has(**"volumeInfo"**)) {  
 JSONObject volumeInfo = item.getJSONObject(**"volumeInfo"**);  
 **if**(volumeInfo.has(**"title"**))  
 title = volumeInfo.getString(**"title"**);  
 **if**(volumeInfo.has(**"subtitle"**))  
 subtitle = volumeInfo.getString(**"subtitle"**);  
 **if**(volumeInfo.has(**"authors"**)) {  
 JSONArray authors = volumeInfo.getJSONArray(**"authors"**);  
 author = authors.getString(0);  
 }  
 **if**(volumeInfo.has(**"imageLinks"**)) {  
 JSONObject imageLinks = volumeInfo.getJSONObject(**"imageLinks"**);  
 **if**(imageLinks.has(**"smallThumbnail"**))  
 imageLink = imageLinks.getString(**"smallThumbnail"**);  
 }  
 }  
  
 **if**(item.has(**"publishedDate"**))  
 publishedDate = item.getString(**"publishedDate"**);  
  
 Book book = **new** Book(title,subtitle,author,publishedDate,imageLink);  
  
 **bookList**.add(book);  
 }  
  
 */\*spinner = (ProgressBar)findViewById(R.id.progressBar1);  
 spinner.setVisibility(View.VISIBLE);\*/* runOnUiThread(**new** Runnable() {  
 @Override  
 **public void** run() {  
  
 **listView** = (RecyclerView) findViewById(R.id.***recyclerview***);  
 **bookAdapter** = **new** ListViewAdapter(**bookList**, getApplication());  
 **listView**.setAdapter(**bookAdapter**);  
 **listView**.setLayoutManager(**new** LinearLayoutManager(**mContext**));  
  
 }  
 });  
  
  
  
 */\* if(spinner!=null){  
  
 spinner.setVisibility(View.GONE);  
 }\*/* }  
  
 }**catch** (Exception e){  
 Log.*d*(**"Async"**,e.getMessage());  
 }  
 }  
 **else** result = **"Did not work!"**;  
  
 } **catch** (Exception e) {  
 Log.*d*(**"InputStream"**, e.getLocalizedMessage());  
 }  
  
  
 **return** result;  
 }  
  
 }  
  
 **private static** String convertInputStreamToString(InputStream inputStream) **throws** IOException{  
 BufferedReader bufferedReader = **new** BufferedReader( **new** InputStreamReader(inputStream));  
 String line = **""**;  
 String result = **""**;  
 **while**((line = bufferedReader.readLine()) != **null**)  
 result += line;  
  
 inputStream.close();  
 **return** result;  
  
 }  
  
 **public void** searchBooks(View view){  
  
 **search** = (EditText) findViewById(R.id.***search***);  
 *//String query = editable.toString().replace(" ", "+");* String query = **"harry+potter"**;  
  
 String getBookURL = **API\_URL**+query;  
  
 **bookList** = **new** ArrayList<Book>();  
  
  
 BookSearchAsyncTaskRunner runnerTask = **new** BookSearchAsyncTaskRunner();  
  
 runnerTask.execute(**"getURL"**,getBookURL);  
 }  
}

List Adapter

**public class** ListViewAdapter **extends** RecyclerView.Adapter<ViewHolder> {  
  
 List<Book> **list** = Collections.*emptyList*();  
 Context **context**;  
  
 **public** ListViewAdapter(List<Book> list, Context context) {  
 **this**.**list** = list;  
 **this**.**context** = context;  
 }  
  
 @Override  
 **public** ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  
  
 View v = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***listlayout***, parent, **false**);  
 ViewHolder holder = **new** ViewHolder(v);  
 **return** holder;  
 }  
  
 @Override  
 **public void** onBindViewHolder(ViewHolder holder, **int** position) {  
  
 holder.**bookTitle**.setText(**list**.get(position).getTitle());  
 holder.**bookSubTitle**.setText(**list**.get(position).getSubtitle());  
 holder.**authors**.setText(**list**.get(position).getAuthors());  
 holder.**publishedDate**.setText(**list**.get(position).getPublishedDate());  
  
 Picasso.*with*(holder.**image**.getContext()).load(**list**.get(position).getImage()).error(android.R.drawable.***sym\_contact\_card***).placeholder(android.R.drawable.***sym\_contact\_card***).into(holder.**image**);  
  
 **final** String bookTitle = **list**.get(position).getTitle();  
 **final** String bookSubTitle = **list**.get(position).getSubtitle();  
 **final** String authors = **list**.get(position).getAuthors();  
 **final** String publishedDate = **list**.get(position).getPublishedDate();  
 **final** String image = **list**.get(position).getImage();  
  
  
 holder.**image**.setOnClickListener(**new** View.OnClickListener(){  
 @Override  
 **public void** onClick(View view){  
  
 Toast.*makeText*(**context**, **"Book Title:"**+bookTitle, Toast.***LENGTH\_SHORT***).show();  
  
 }  
 });  
  
 }  
  
  
  
}

View Holder:

**public class** ViewHolder **extends** RecyclerView.ViewHolder{  
  
 **public** CardView **cv**;  
 **public** TextView **bookTitle**;  
 **public** TextView **bookSubTitle**;  
 **public** TextView **publishedDate**;  
 **public** TextView **authors**;  
 **public** ImageView **image**;  
  
 **public** ViewHolder(View itemView) {  
 **super**(itemView);  
 **cv** = (CardView) itemView.findViewById(R.id.***cardView***);  
 **bookTitle** = (TextView) itemView.findViewById(R.id.***bookTitle***);  
 **bookSubTitle** = (TextView) itemView.findViewById(R.id.***bookSubTitle***);  
 **publishedDate** = (TextView) itemView.findViewById(R.id.***publishedDate***);  
 **authors** = (TextView) itemView.findViewById(R.id.***authors***);  
 **image** = (ImageView) itemView.findViewById(R.id.***image***);  
 }  
  
}

Book class:

**public class** Book {  
  
 **private** String **title**;  
  
 **private** String **subtitle**;  
  
 **private** String **authors**;  
  
 **private** String **publishedDate**;  
  
 **private** String **image**;  
  
 Book(String title, String subtitle, String authors, String publishedDate, String image){  
 **this**.**title** = title;  
 **this**.**subtitle** = subtitle;  
 **this**.**authors** = authors;  
 **this**.**publishedDate** = publishedDate;  
 **this**.**image** = image;  
 }  
  
 **public** String getTitle() {  
 **return title**;  
 }  
  
 **public void** setTitle(String title) {  
 **this**.**title** = title;  
 }  
  
 **public** String getSubtitle() {  
 **return subtitle**;  
 }  
  
 **public void** setSubtitle(String subtitle) {  
 **this**.**subtitle** = subtitle;  
 }  
  
 **public** String getAuthors() {  
 **return authors**;  
 }  
  
 **public void** setAuthors(String authors) {  
 **this**.**authors** = authors;  
 }  
  
 **public** String getPublishedDate() {  
 **return publishedDate**;  
 }  
  
 **public void** setPublishedDate(String publishedDate) {  
 **this**.**publishedDate** = publishedDate;  
 }  
  
 **public** String getImage() {  
 **return image**;  
 }  
  
 **public void** setImage(String image) {  
 **this**.**image** = image;  
 }  
}

activity index:

<**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**include layout="@layout/listcontent"** />

</**RelativeLayout**>

List content:

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context=".IndexActivity"  
 tools:showIn="@layout/activity\_index"**>  
  
 <**EditText android:id="@+id/search"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Search books.."  
 android:layout\_marginTop="100dp"  
 android:textColorHighlight="@android:color/white"  
 android:textColorHint="@android:color/white"  
 android:textColorLink="@android:color/white"  
 android:textSize="20dp"  
 android:textColor="@android:color/white"**/>  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/searchBtn"  
 android:layout\_below="@+id/search"  
 android:onClick="searchBooks"  
 android:text="Search"**/>  
  
 <**android.support.v7.widget.RecyclerView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/recyclerview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/search"**/>  
  
  
</**RelativeLayout**>

List Layout:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"**>  
  
 <**android.support.v7.widget.CardView  
 android:id="@+id/cardView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 app:cardCornerRadius="12dp"  
 app:cardElevation="6dp"  
 app:cardBackgroundColor="@android:color/transparent"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:padding="9dp"  
 android:layout\_height="190dp"**>  
  
 <**ImageView  
 android:id="@+id/image"  
 android:layout\_width="90dp"  
 android:layout\_height="230dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginRight="9dp"** />  
  
 <**TextView  
 android:id="@+id/bookTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_toRightOf="@+id/image"  
 android:text="Title"  
 android:textSize="18dp"  
 android:paddingTop="15dp"  
 android:textColor="@android:color/white"**/>  
  
 <**TextView  
 android:id="@+id/bookSubTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/bookTitle"  
 android:layout\_toRightOf="@+id/image"  
 android:text="Title"  
 android:textSize="12dp"  
 android:paddingTop="10dp"  
 android:textColor="@android:color/white"**/>  
  
 <**TextView  
 android:id="@+id/authors"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/bookSubTitle"  
 android:layout\_toRightOf="@+id/image"  
 android:text="Description"  
 android:textSize="13dp"  
 android:paddingTop="10dp"  
 android:textColor="@android:color/white"**/>  
  
 <**TextView  
 android:id="@+id/publishedDate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/authors"  
 android:layout\_toRightOf="@+id/image"  
 android:text="Published Date"  
 android:textSize="13dp"  
 android:paddingTop="10dp"  
 android:textColor="@android:color/white"**/>  
  
  
 </**RelativeLayout**>  
  
 </**android.support.v7.widget.CardView**>  
  
  
  
</**LinearLayout**>

1. Guardian API in Grid View:

Activity:

**public class** HelpActivity **extends** AppCompatActivity  
 **implements** NavigationView.OnNavigationItemSelectedListener {  
  
 String **sharedText**;  
 **private** String[] **mThumbIds**;  
 **private** Context **mContext**;  
  
 String **API\_URL** = **"http://content.guardianapis.com/search?q=debates&api-key=test&show-fields=thumbnail"**;  
  
 **public** HelpActivity(){  
 **super**();  
  
 }  
 **public** HelpActivity(Context context){  
  
 **mContext** = context;  
 }  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_help***);  
 **mContext** = getApplicationContext();  
   
 Intent intent = getIntent();  
 String action = intent.getAction();  
 String type = intent.getType();  
  
 **if** (Intent.***ACTION\_SEND***.equals(action) && type != **null**) {  
 **if** (**"text/plain"**.equals(type)) {  
 handleSendText(intent); *// Handle text being sent* }  
 }  
  
 }  
  
 **void** handleSendText(Intent intent) {  
 GuardianAsyncTaskRunner runnerTask = **new** GuardianAsyncTaskRunner();  
  
 runnerTask.execute(**"getURL"**,**API\_URL**);  
 }  
  
 **private class** GuardianAsyncTaskRunner **extends** AsyncTask<String, String, String> {  
  
 **private** String **resp**;  
 ProgressBar **spinner**;  
  
 @Override  
 **protected** String doInBackground(String... strings) {  
  
 InputStream inputStream = **null**;  
 String result = **""**;  
 **try** {  
  
 *// create HttpClient* HttpClient httpclient = **new** DefaultHttpClient();  
  
 *// make GET request to the given URL* HttpResponse httpResponse = httpclient.execute(**new** HttpGet(strings[1]));  
  
 *// receive response as inputStream* inputStream = httpResponse.getEntity().getContent();  
  
 *// convert inputstream to string* **if**(inputStream != **null**) {  
 result = *convertInputStreamToString*(inputStream);  
 **try** {  
 **if**(result!=**null** && !**""**.equalsIgnoreCase(result)){  
  
 JSONObject jsonResponse = **new** JSONObject(result);  
 JSONObject jsonResults = jsonResponse.getJSONObject(**"response"**);  
 JSONArray resultsArray = jsonResults.getJSONArray(**"results"**);  
 List<String> imgList = **new** ArrayList<String>();  
 **for** (**int** i = 0; i <=7; i++) {  
  
 JSONObject item = resultsArray.getJSONObject(i);  
  
 String img = **""**;  
  
 **if**(item.has(**"fields"**)) {  
 JSONObject fieldsInfo = item.getJSONObject(**"fields"**);  
 **if**(fieldsInfo.has(**"thumbnail"**))  
 img = fieldsInfo.getString(**"thumbnail"**);  
 }  
  
 imgList.add(img);  
 }  
 **if**(imgList.size()>0)  
 **mThumbIds** = **new** String[]{  
 imgList.get(0), imgList.get(1),  
 imgList.get(2), imgList.get(3),  
 imgList.get(4), imgList.get(5),  
 imgList.get(6), imgList.get(7)  
 };  
  
 runOnUiThread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 GridView gridview = (GridView) findViewById(R.id.***gridview***);  
 gridview.setAdapter(**new** GridAdapter(**mContext**, **mThumbIds**));  
  
 gridview.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 **public void** onItemClick(AdapterView<?> parent, View v,  
 **int** position, **long** id) {  
 Toast.*makeText*(HelpActivity.**this**, **""** + position,  
 Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
  
  
  
 }  
 });  
  
 }  
  
 }**catch** (Exception e){  
 Log.*d*(**"Async"**,e.getMessage());  
 }  
 }  
 **else** result = **"Did not work!"**;  
  
 } **catch** (Exception e) {  
 Log.*d*(**"InputStream"**, e.getLocalizedMessage());  
 }  
  
  
 **return** result;  
 }  
  
 }  
  
 **private static** String convertInputStreamToString(InputStream inputStream) **throws** IOException {  
 BufferedReader bufferedReader = **new** BufferedReader( **new** InputStreamReader(inputStream));  
 String line = **""**;  
 String result = **""**;  
 **while**((line = bufferedReader.readLine()) != **null**)  
 result += line;  
  
 inputStream.close();  
 **return** result;  
  
 }  
  
}

Grid Adapter:

**public class** GridAdapter **extends** BaseAdapter {  
  
 **private** Context **mContext**;  
 **private** String[] **mThumbIds**;  
  
 **public** GridAdapter(Context c,String[] mThumbIds) {  
 **mContext** = c;  
 **this**.**mThumbIds** = mThumbIds;  
  
 }  
  
 @Override  
 **public int** getCount() {  
 **return mThumbIds**.**length**;  
 }  
 @Override  
 **public** String getItem(**int** position) {  
 **return mThumbIds**[position];  
 }  
  
 **public long** getItemId(**int** position) {  
 **return** 0;  
 }  
  
 ArrayList<String> **guardianList** = **new** ArrayList<String>();  
  
 *// create a new ImageView for each item referenced by the Adapter* **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 ImageView imageView;  
 **if** (convertView == **null**) {  
 imageView = **new** ImageView(**mContext**);  
 imageView.setLayoutParams(**new** GridView.LayoutParams(200, 200));  
 imageView.setScaleType(ImageView.ScaleType.***CENTER\_CROP***);  
 imageView.setPadding(8, 8, 8, 8);  
 } **else** {  
 imageView = (ImageView) convertView;  
 }  
  
 String url = getItem(position);  
 Picasso.*with*(**mContext**)  
 .load(url).into(imageView);  
  
 **return** imageView;  
 }  
}

layout:

<**RelativeLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"**>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Latest Debate News"  
 android:textSize="20dp"  
 android:layout\_marginTop="100dp"  
 android:textColor="@android:color/background\_light"  
 android:id="@+id/text"**/>  
 <**GridView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/gridview"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:columnWidth="90dp"  
 android:numColumns="auto\_fit"  
 android:verticalSpacing="10dp"  
 android:horizontalSpacing="10dp"  
 android:stretchMode="columnWidth"  
 android:gravity="center"  
 android:layout\_below="@+id/text"** />  
</**RelativeLayout**>

Gradle file included:

compile **'com.android.support:cardview-v7:26.+'**compile **'com.android.support:recyclerview-v7:26.+'**compile **'com.squareup.picasso:picasso:2.5.2'**

useLibrary **'org.apache.http.legacy'**