Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

eSkill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation

ANDROID APPLICATION DEVELOPMENT

ASYNC TASK

AsyncTask

Introduction:

A thread is an independent path of execution in a running program. When an Android program is launched, the Android Runtime system creates a thread called the "Main" thread. As your program runs, each line of code is executed in a serial fashion, line by line. This main thread is how your application interacts with components from the Android UI Toolkit, and it's why the main thread is sometimes called the "UI thread". However, sometimes an application needs to perform resource-intensive work, such as downloading files, database queries, playing media, or computing complex analytics. This type of intensive work can block the UI thread if all of the code executes serially on a single thread. When the app is performing resources intensive work, the app does not respond to the user or draw on the screen because it is waiting for that work to be done. This can yield poor performance, which negatively affects the user experience. Users may get frustrated and uninstall your Android app if the performance of the app is slow.

To keep the user experience (UX) running smoothly and responding quickly to user gestures, the Android Framework provides a helper class called AsyncTask which processes work off of the UI thread. An AsyncTask is an abstract Java class that provides one way to move this intensive processing onto a separate thread, thereby allowing the UI thread to remain very responsive. Since the separate thread is not synchronized with the calling thread, it is called an asynchronous thread. An AsyncTask also contains a callback that allows you to display the results of the computation back in the UI thread.

In this practical, you will learn how to add a background task to your Android app using an AsyncTask.

Async-task for Android

This library does the same thing AsyncTask does – runs code on background thread and gets result back to the main thread. But it is simpler, cleaner, easier to understand, straightforward to use, requires no arcane knowledge to be used. And it requires less boilerplate code than RxJava.

Methods of AsyncTask

onPreExecute()

Before doing background operation we should show something on screen like progress bar or any animation to user. we can directly communicate background operation using on doInBackground() but for the best practice, we should call all asyncTask methods.

doInBackground(Params)

In this method we have to do background operation on background thread. Operations in this method should not touch on any main thread activities or fragments.

onProgressUpdate(Progress…)

While doing background operation, if you want to update some information on UI, we can use this method.

onPostExecute(Result)

In this method we can update ui of background operation result.
Generic Types in Async Task

TypeOfVarArgParams

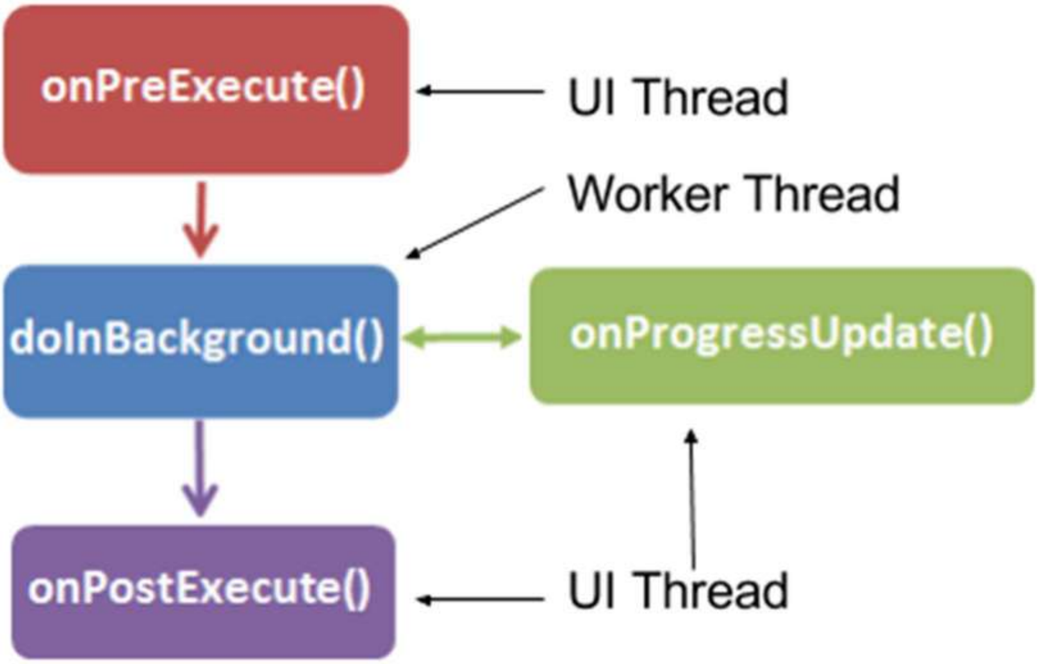
It contains information about what type of params used for execution.

ProgressValue

It contains information about progress units. While doing background operation we can update information on ui using onProgressUpdate().

ResultValue

It contains information about result type.



Sample Asyntask Example Application

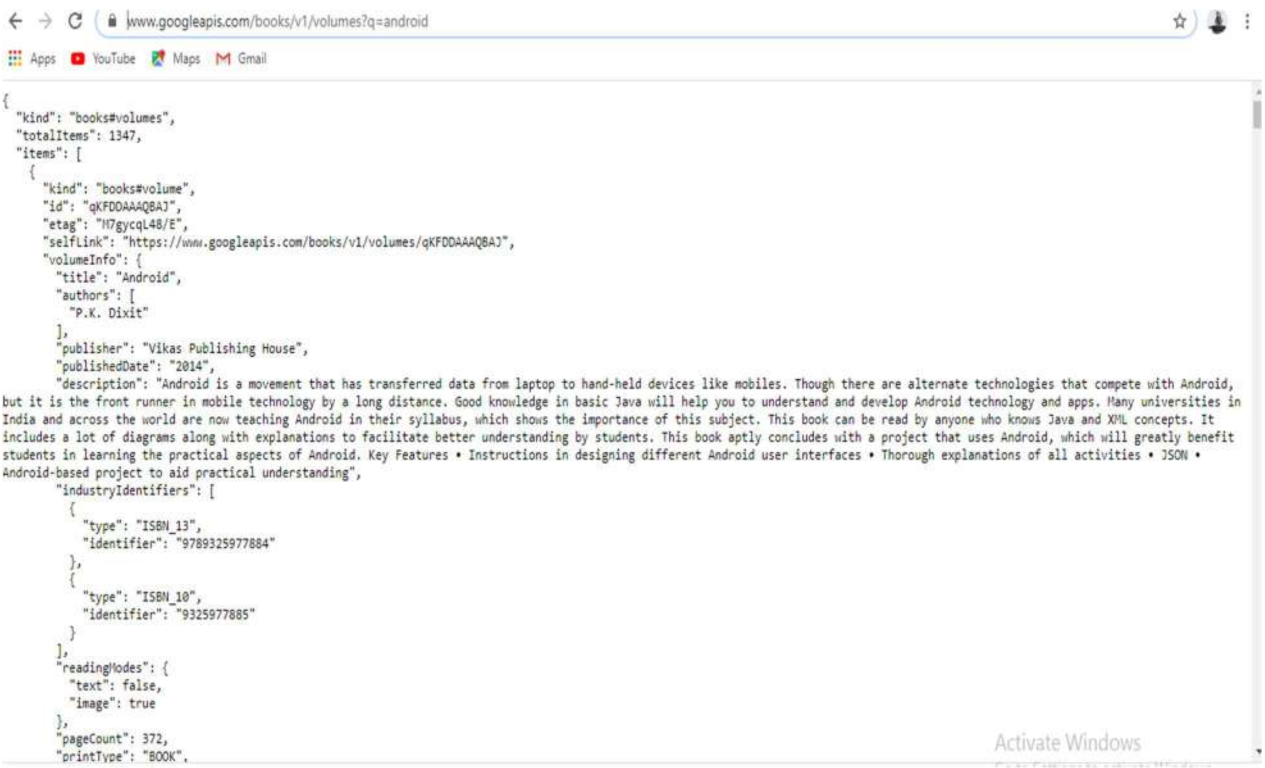
Step1:

Add the Recyclerview Dependency at build.gradle(Module:app)

implementation 'com.android.support:recyclerview-v7:28.0.0'

Step2

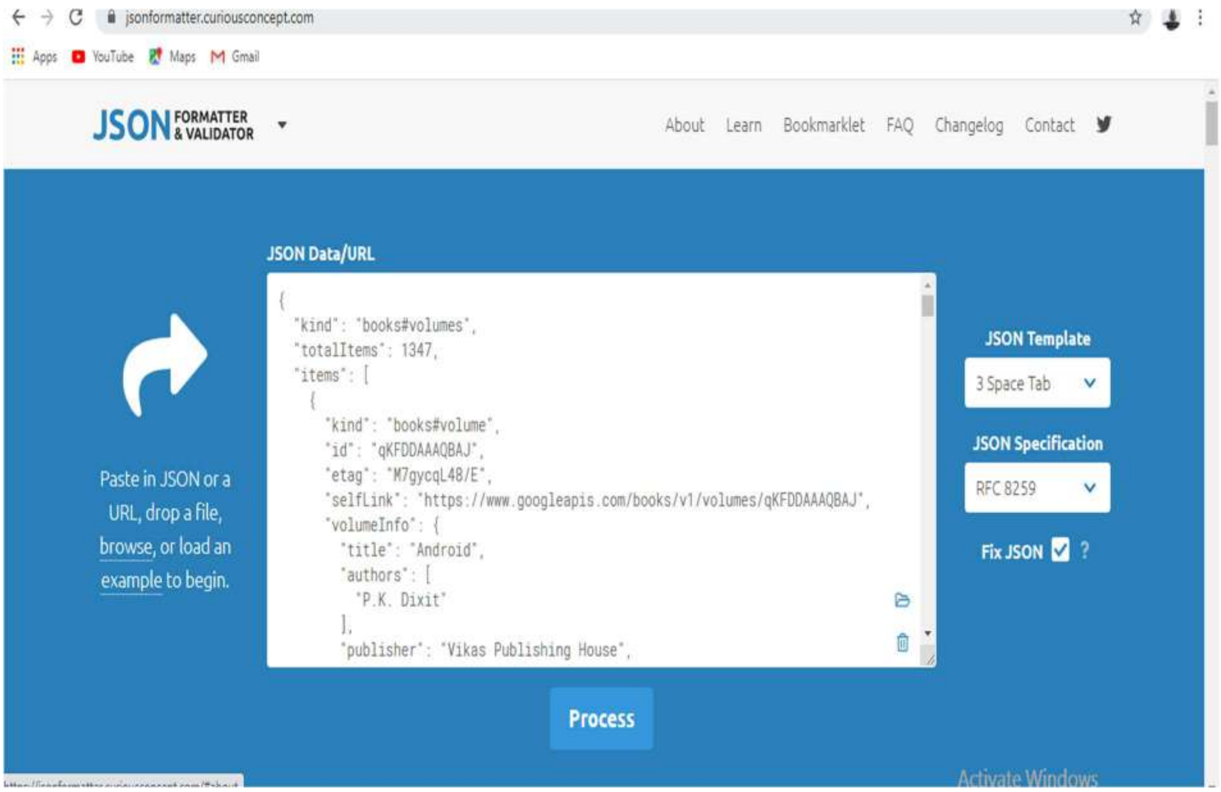
Api link ="https://www.googleapis.com/books/v1/volumes?q=android" acces the url



```
{
  "kind": "books#volumes",
  "totalItems": 1347,
  "items": [
    {
      "kind": "books#volume",
      "id": "qKFDDAAAQBAJ",
      "etag": "M7gycqL48/E",
      "selfLink": "https://www.googleapis.com/books/v1/volumes/qKFDDAAAQBAJ",
      "volumeInfo": {
        "title": "Android",
        "authors": [
          "P.K. Dixit"
        ],
        "publisher": "Vikas Publishing House",
        "publishedDate": "2014",
        "description": "Android is a movement that has transferred data from laptop to hand-held devices like mobiles. Though there are alternate technologies that compete with Android,
but it is the front runner in mobile technology by a long distance. Good knowledge in basic Java will help you to understand and develop Android technology and apps. Many universities in
India and across the world are now teaching Android in their syllabus, which shows the importance of this subject. This book can be read by anyone who knows Java and XML concepts. It
includes a lot of diagrams along with explanations to facilitate better understanding by students. This book aptly concludes with a project that uses Android, which will greatly benefit
students in learning the practical aspects of Android. Key Features • Instructions in designing different Android user interfaces • Thorough explanations of all activities • JSON •
Android-based project to aid practical understanding",
        "industryIdentifiers": [
          {
            "type": "ISBN_13",
            "identifier": "9789325977884"
          },
          {
            "type": "ISBN_10",
            "identifier": "9325977885"
          }
        ],
        "readingModes": {
          "text": false,
          "image": true
        },
        "pageCount": 372,
        "printType": "BOOK",
```

Step3

Copy the url data into a json formatter website.



jsonformatter.curiousconcept.com
Apps
YouTube
Maps
Gmail
JSON FORMATTER & VALIDATOR
About
Learn
Bookmarklet
FAQ
Changelog
Contact
JSON Data/URL
Paste in JSON or a URL, drop a file, browse, or load an example to begin.
JSON Template
JSON Specification
Process
Activate Windows

JSON Template: 3 Space Tab

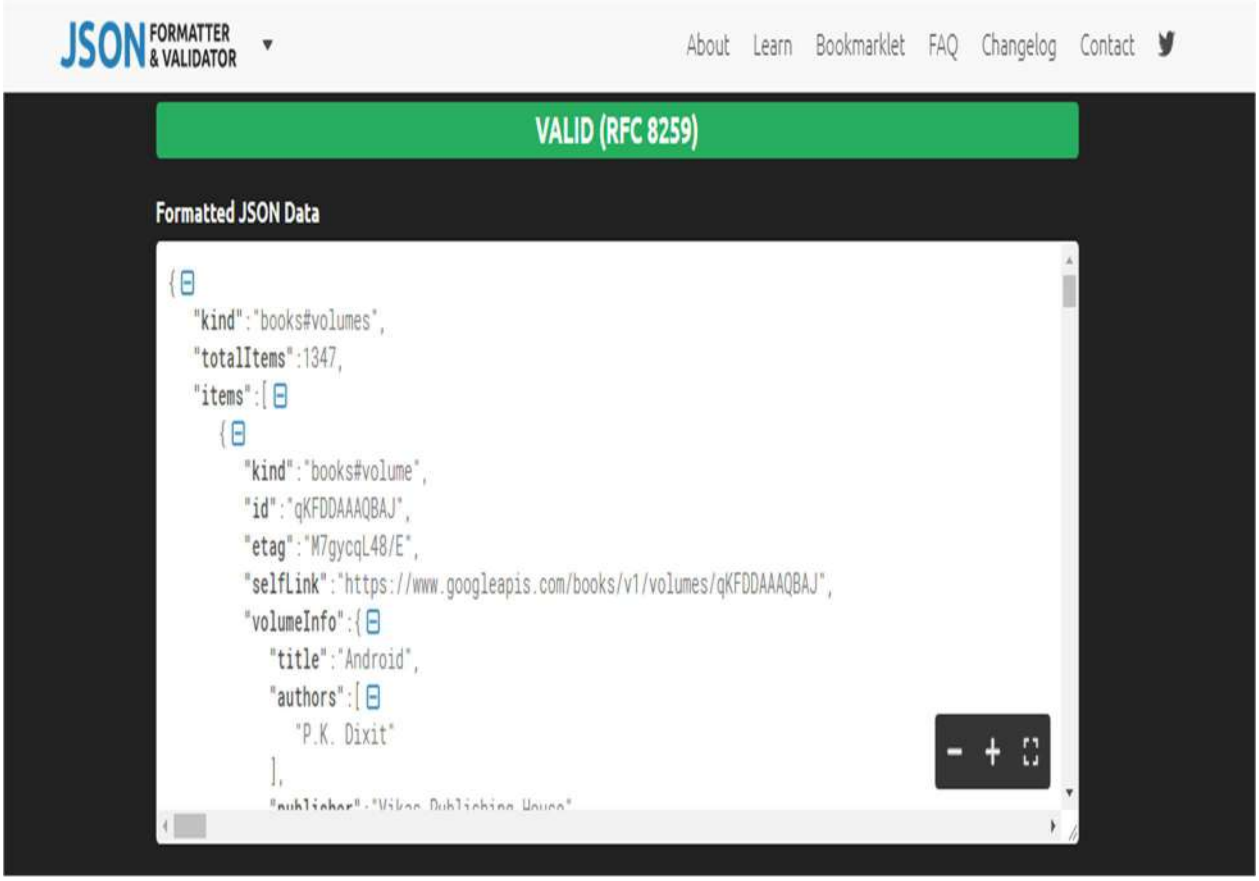
JSON Specification: RFC 8259

Fix JSON ☑ ?

```
{
  "kind": "books#volumes",
  "totalItems": 1347,
  "items": [
    {
      "kind": "books#volume",
      "id": "qKFDDAAAQBAJ",
      "etag": "M7gycqL48/E",
      "selfLink": "https://www.googleapis.com/books/v1/volumes/qKFDDAAAQBAJ",
      "volumeInfo": {
        "title": "Android",
        "authors": [
          "P.K. Dixit"
        ],
        "publisher": "Vikas Publishing House",
```

Step4

check given json data is valid are not.



VALID (RFC 8259)

Formatted JSON Data

```
{
   "kind":"books#volumes",
   "totalItems":1347,
   "items":[
      {
         "kind":"books#volume",
         "id":"qKFDDAAAQBAJ",
         "etag":"M7gycqL48/E",
         "selfLink":"https://www.googleapis.com/books/v1/volumes/qKFDDAAAQBAJ",
         "volumeInfo":{
            "title":"Android",
            "authors":[
               "P.K. Dixit"
            ],
            "publisher":"Vikas Publishing House",
```

Step5

Create the Array List Class Name As Book.

```
package com.example.acer.gopalbook;

public class Book
{
   String title;
   String sub;
   String author;
   String img;
   public Book() {
      this.title = title;
      this.sub = sub;
      this.author = author;
      this.img = img;
   }
   public String getTitle() {
      return title;
   }

   public void setTitle(String title) {
      this.title = title;
   }

   public String getSub() {
      return sub;
   }

   public void setSub(String sub) {
      this.sub = sub;
   }

   public String getAuthor() {
      return author;
   }

   public void setAuthor(String author) {
      this.author = author;
   }

   public String getImg() {
      return img;
   }

   public void setImg(String img) {
      this.img = img;
   }


}
```

Step6

create an AsynTask Class Name is MyTask extends AsyncTask

```
package com.example.acer.gopalbook;

import android.app.ProgressDialog;
import android.content.Context;
import android.net.Uri;
import android.os.AsyncTask;
import android.support.v7.widget.DividerItemDecoration;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;

class MyTask  extends AsyncTask<Void,String,String> {
    Context context;
    int position;
    ProgressDialog pd;
    String BASE_URL="https://www.googleapis.com/books/v1/volumes?q=android";
    RecyclerView rv;

    public MyTask(Context context,RecyclerView rv,int position) {
        this.context = context;
        this.rv=rv;
        this.position=position;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pd=new ProgressDialog(context);
        pd.setTitle("Please wait");
        pd.setMessage("Books info is loading");
        pd.show();
    }

    @Override
    protected String doInBackground(Void... voids) {
        Uri uri=Uri.parse(BASE_URL);
        try {
            URL url=new URL(uri.toString());
            HttpURLConnection connection= (HttpURLConnection) url.openConnection();
            connection.connect();
            InputStream inputStream=connection.getInputStream();
            BufferedReader bufferedReader=new BufferedReader(new InputStreamReader(inputStre
am));
            String temp="";
            StringBuilder sb=new StringBuilder();
            while ((temp=bufferedReader.readLine())!=null){
                sb.append(temp);
            }
            connection.disconnect();
            return sb.toString();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        pd.dismiss();
        String image[]=new String[10];
        String title[]=new String[10];
        String authors[]=new String[10];
        String publisher[]=new String[10];
        String date[]=new String[10];
        String descr[]=new String[10];
        try {
            JSONObject root=new JSONObject(s);
            JSONArray items=root.getJSONArray("items");
            authors[0]="";
            for(int i=0;i<10;i++){
                JSONObject info=items.getJSONObject(i);
                JSONObject volume=info.getJSONObject("volumeInfo");
                String temp=volume.getString("title");
                title[i]=temp;
                temp=volume.optString("description");
                descr[i]=temp;
                temp=volume.optString("publisher");
                publisher[i]=temp;
                temp=volume.optString("publishedDate");
                date[i]=temp;
                JSONArray auth=volume.getJSONArray("authors");
                temp="";
                for (int j=0;j<auth.length();j++){
                    temp=temp+auth.getString(j)+",";
                    authors[i]=temp;
                }
                JSONObject img=volume.getJSONObject("imageLinks");
                temp=img.getString("thumbnail");
                image[i]=temp;
            }
            MyAdapter adapter=new MyAdapter(context,image,title,authors,date,descr,publisher);
            rv.setAdapter(adapter);
            rv.setLayoutManager(new LinearLayoutManager(context));
            rv.scrollToPosition(position);
            rv.addItemDecoration(new DividerItemDecoration(context,DividerItemDecoration.VERT
ICAL));
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

Step 7

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    android:orientation="vertical"
    tools:context=".MainActivity">


<android.support.v7.widget.RecyclerView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/rec">


</android.support.v7.widget.RecyclerView>


</LinearLayout>
```

Step8

MainActivity

```
package com.example.acer.gopalbook;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    RecyclerView recyclerView;

    EditText book;
    static int position=0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        recyclerView=findViewById(R.id.rec);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        recyclerView.setHasFixedSize(true);
        MyTask task=new MyTask(this,recyclerView,position);
        task.execute();
    }
}
```

Step 9

row.xml file

```
<?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="wrap_content"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ImageView
            android:id="@+id/img_books"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:scaleType="fitXY"/>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <TextView
                android:id="@+id/tv_title"
                android:text="title"
                android:textSize="25sp"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/tv_authors"
                android:textSize="25sp"
                android:text="authors"/>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

Step10

MyAdapter Class

```
package com.example.acer.gopalbook;

import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.squareup.picasso.Picasso;

import java.util.ArrayList;

class MyAdapter extends RecyclerView.Adapter<MyAdapter.BooksInfo> {
    Context context;
    String image_url[];
    String title[];
    String authors[];
    String date[];
    String description[];
    String publisher[];

    public MyAdapter(Context context,String image[], String title[], String[] authors, String[] date
, String[] description, String[] publisher) {
        this.context = context;
        this.title = title;
        this.authors = authors;
        this.date = date;
        this.description = description;
        this.publisher = publisher;
        this.image_url=image;
    }

    @NonNull
    @Override
    public BooksInfo onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view= LayoutInflater.from(context).inflate(R.layout.row,parent,false);
        return new BooksInfo(view);
    }

    @Override
    public void onBindViewHolder(@NonNull BooksInfo holder, final int position) {
        Picasso.with(context).load(image_url[position]).into(holder.img);
        holder.tv1.setText(""+title[position]);
        holder.tv2.setText(""+authors[position]);

    }

    @Override
    public int getItemCount() {
        //if(title!=null)
        return title.length;
        //else
        //  return 0;
    }

    public class BooksInfo extends RecyclerView.ViewHolder {
        ImageView img;
        TextView tv1,tv2;

        public BooksInfo(View itemView) {
            super(itemView);
            img=itemView.findViewById(R.id.img_books);
            tv1=itemView.findViewById(R.id.tv_title);
            tv2=itemView.findViewById(R.id.tv_authors);
        }
    }
}
```