

How To's

Linked How-To's:

Decrease Loading time for app with background image:

For this app one of the main things I wanted to have my app stand out from the others is the UI and Graphics. I wanted super clean simple graphics. But when I tried to upload my image to be the background for the main activity, the functions of the actual app would take forever to load. I tried taking off the background image and the app worked perfectly fine. The solution I found was to import the image in all the drawable folders, and I downloaded a plugin to assist with that later!

Link:

<https://stackoverflow.com/questions/17335874/slow-app-with-background-image>

Plugin:

<https://github.com/winterDroid/android-drawable-importer-intellij-plugin>

How to install the Glide Library:

Like I said in the previous How-To, for my app I want clean graphics and I thought playing GIFs in the app as icons would be a good idea to distinguish myself from other apps. So I googled libraries to help do so and I came across Glide! But I had no experience utilizing libraries so I went on the StackOverflow to figure out how to install and utilize it.

Link:

<https://stackoverflow.com/questions/25610727/adding-external-library-in-an-droid-studio>

Detailed How-To;

Combining ListViews and Class Objects and Having a Seperate Layout for Different Orientations:

Description:

In my app I wanted to use the NASA Web API to list out the asteroids passing by Earth in a given date range, so utilizing the API gives a long list of asteroids. I have attached to data gotten from the API below.

```
{
  "links": {
    "next": "https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-08&end_date=2015-09-08&detailed=false&api_key=Tel10Gw",
    "prev": "https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-06&end_date=2015-09-07&detailed=false&api_key=Tel10Gw",
    "self": "https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-07&end_date=2015-09-08&detailed=false&api_key=Tel10Gw"
  },
  "element_count": 21,
  "near_earth_objects": {
    "2015-09-08": [ {
      "links": {
        "self": "https://api.nasa.gov/neo/rest/v1/neo/3726710?api_key=Tel10Gw#UXnnnIXFV19ICT7G2E446Lp3du188u"
      },
      "neo_reference_id": "3726710",
      "name": "(2015 RC)",
      "nasa_jpl_url": "http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3726710",
      "absolute_magnitude_h": 24.3,
      "estimated_diameter": {
        "kilometers": {
          "estimated_diameter_min": 0.0366906138,
          "estimated_diameter_max": 0.0820427065
        },
        "meters": {
          "estimated_diameter_min": 36.6906137531,
          "estimated_diameter_max": 82.0427064882
        },
        "miles": {
          "estimated_diameter_min": 0.0227984834,
          "estimated_diameter_max": 0.0509789586
        },
        "feet": {
          "estimated_diameter_min": 120.3760332259,
          "estimated_diameter_max": 269.1689931548
        }
      },
      "is_potentially_hazardous_asteroid": false,
      "close_approach_data": [ {
        "close_approach_date": "2015-09-08",
        "epoch_date_close_approach": 1441695000000,
        "relative_velocity": {
          "kilometers_per_second": "19.4850295284",
          "kilometers_per_hour": "70146.106302123",
          "miles_per_hour": "43586.0625520053"
        },
        "miss_distance": {
          "astronomical": "0.0269230459",
          "lunar": "10.4730644226",
          "kilometers": "4027630.25",
          "miles": "2502653.5"
        },
        "orbiting_body": "Earth"
      } ]
    }, {
      "links": {
        "self": "https://api.nasa.gov/neo/rest/v1/neo/2465633?api_key=Tel10Gw#UXnnnIXFV19ICT7G2E446Lp3du188u"
      },
      "neo_reference_id": "2465633",
      "name": "465633 (2009 JH5)",
      "nasa_jpl_url": "http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=2465633",
      "absolute_magnitude_h": 20.3,
      "estimated_diameter": {
        "kilometers": {
          "estimated_diameter_min": 0.2315021222,

```

Asteroid Class Object

I wanted to make an ArrayList of Asteroid Class Objects that I defined in my program to organize and sort all the information. And for one of the activities, I wanted to display this objects in a ListView for the Landscape orientation but was super confused how to implement an ArrayList of class objects in that. And a LinearLayout /RelativeLayout in the Portrait mode.

```
public class Asteroid {
    public String ID;
    public String name;
    public String minDiameter;
    public String maxDiameter;
    public String potentialHazard;
    public String closeApproachDate;
    public String velocity;
    public String missDistance;
    public int index;
    private ArrayList<Asteroid> list;

    public Asteroid(int Index, String numID, String Name, String mDiameter, String mxDiameter,

        ID = numID;
        name = Name;
        minDiameter = mDiameter;
        maxDiameter = mxDiameter;
        potentialHazard = hazard;
        closeApproachDate = cAdate;
        velocity = velo;
        missDistance = mDistance;
        index = Index;
    }

    public String getName() {
        String rawLine = this.name;
        String[] lineSegments = rawLine.split( regex: "\\s+", limit: 3);
        String finalName = lineSegments[2].substring(3, lineSegments[2].length() - 2);

        return finalName;
    }

    public String getID() {
        String rawLine = this.ID;
        String[] lineSegments = rawLine.split( regex: "\\s+", limit: 3);
        String ID = lineSegments[2];
        String words = ID.replaceAll( regex: "\\p{P}", replacement: "");
        return words;
    }
}
```

Solution:

1. First I wanted to start on the ListView and I needed to create an XML file for the specific ListView item to structure it, pretty simple relativeLayout to display two TextViews (I am going to add more elements to this layout later)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- <ImageView
        android:id="@+id/imageView_poster"
        android:layout_width="150dp"
        android:layout_height="200dp"/>-->

    <TextView
        android:id="@+id/textView_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:textSize="20sp"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/textView_release"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/textView_name"
        android:layout_margin="20dp"
        android:textSize="20sp"/>

</RelativeLayout>
```

2. Next I learned that I needed to create an ListAdapter to completely control the list view and assign values to them. This adapter lets me import the context from the main activity and the asteroid ArrayList and overrides the View with the List item elements and also inherits the XML we defined earlier. Then I assigned the specific TextViews with data from the asteroid ArrayList I imported from the main Activity.

```
public class AsteroidAdapter extends ArrayAdapter<Asteroid> {  
  
    private Context mContext;  
    NasaDataActivity mData = new NasaDataActivity();  
    ArrayList<Asteroid> asteroidList = mData.asteroidGroup;  
  
    public AsteroidAdapter(@NonNull Context context, ArrayList<Asteroid> list) {  
        super(context, resource 0, list);  
        mContext = context;  
        asteroidList = list;  
    }  
  
    @Override  
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
        View listItem = convertView;  
        if(listItem == null)  
            listItem = LayoutInflater.from(mContext).inflate(R.layout.list_item,parent, attachToRoot: false);  
  
        Asteroid currentAsteroid = asteroidList.get(position);  
        TextView name = (TextView) listItem.findViewById(R.id.textView_name);  
        name.setText(currentAsteroid.getName());  
  
        TextView release = (TextView) listItem.findViewById(R.id.textView_release);  
        release.setText(currentAsteroid.getID());  
  
        return listItem;  
    }  
}
```

3. Then in the main Activity I had to pull the Asteroid List from the my Data Retrieving Class named “NasaDataActivity” and instantiate a ListView in the if statement for checking the orientation of the phone. I then instantiate an adapter with the asteroid List and context to override the View and and create a OnClickListener to listen for button clicks for each specific object. Which will lead to another activity displaying the information for the Asteroid. This completed the ListView layout

```
setContentView(R.layout.activity_asteroid_list);  
ListView lv = findViewById(android.R.id.list);  
if (lv != null) {  
    AsteroidAdapter mAdapter;  
    mAdapter = new AsteroidAdapter( context: this, asteroidList);  
    lv.setAdapter(mAdapter);  
    lv.setOnItemClickListener((parent, view, position, id) -> {  
  
        // Retrieve our class object and use index to resolve item tapped  
        final Asteroid item = asteroidList.get(position);  
        final int menuIndex = item.getIndex();  
        switch (menuIndex) {  
            case 1:  
                Log.d( tag: "Test", msg: "Hi");  
                break;  
            case 2:  
                break;  
            case 3:  
                break;  
            default:  
                break;  
        }  
    });  
}
```

4. The hard part was to create two different Layouts for the two orientations, we went over it in one of the tutorials but it used the same type of Layout I believe. I tried to create the two different XMLS layouts for the activity, one being specifically for Landscape, as we did in the the tutorial but it caused the app to crash because I did not include that if statement and set the Content View immediately when onCreate was called. I also instantiate a ListView in the beginning, which caused a crash if the phone was in portrait mode, but would work fine in landscape mode. I also tried setting if statements to see if the ListView is null and if so to do the other layout, but that also did not work. I then looked up and found out someone can set the ContentView in if statements and I looked up how to find the orientation of the phone which was a struggle in itself because many of the online resources said i could use

```
int currentOrientation = getResources().getConfiguration().orientation;
```

And it would not give me a syntax error, but the app would crash every run now and I could not figure out why. Doing more research I found out that the correct usage of this is

```
int currentOrientation = this.getResources().getConfiguration().orientation;
```

After figuring this out I just set up a simple if statement and set the ContentView in each of them. Finished Main Activity code below

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    NASADataActivity mData = new NASADataActivity();
    final ArrayList<Asteroid> asteroidList = mData.asteroidGroup;

    int currentOrientation = this.getResources().getConfiguration().orientation;

    if (currentOrientation == Configuration.ORIENTATION_LANDSCAPE) {
        setContentView(R.layout.activity_asteroid_list);
        ListView lv = findViewById(android.R.id.list);
        if (lv != null) {
            AsteroidAdapter mAdapter;
            mAdapter = new AsteroidAdapter( context: this, asteroidList);
            lv.setAdapter(mAdapter);
            lv.setOnItemClickListener((parent, view, position, id) -> {

                // Retrieve our class object and use index to resolve item tapped
                final Asteroid item = asteroidList.get(position);
                final int menuIndex = item.getIndex();
                switch (menuIndex) {
                    case 1:
                        Log.d( tag: "Test", msg: "Hi");
                        break;
                    case 2:
                        break;
                    case 3:
                        break;
                    default:
                        break;
                }
            });
        }
    }

    if (currentOrientation == Configuration.ORIENTATION_PORTRAIT) {
        setContentView(R.layout.activity_asteroid_list);
        TextView asteroidName = (TextView) findViewById(R.id.asteroidName);
        TextView passbyDate = (TextView) findViewById(R.id.passByDate);
        asteroidName.setText(asteroidList.get(0).getName());
        passbyDate.setText(asteroidList.get(0).getCloseApproachDate());
    }
}

```