

# Android Workshop

Luke Klinker  
Source Allies Inc.

# Introduction

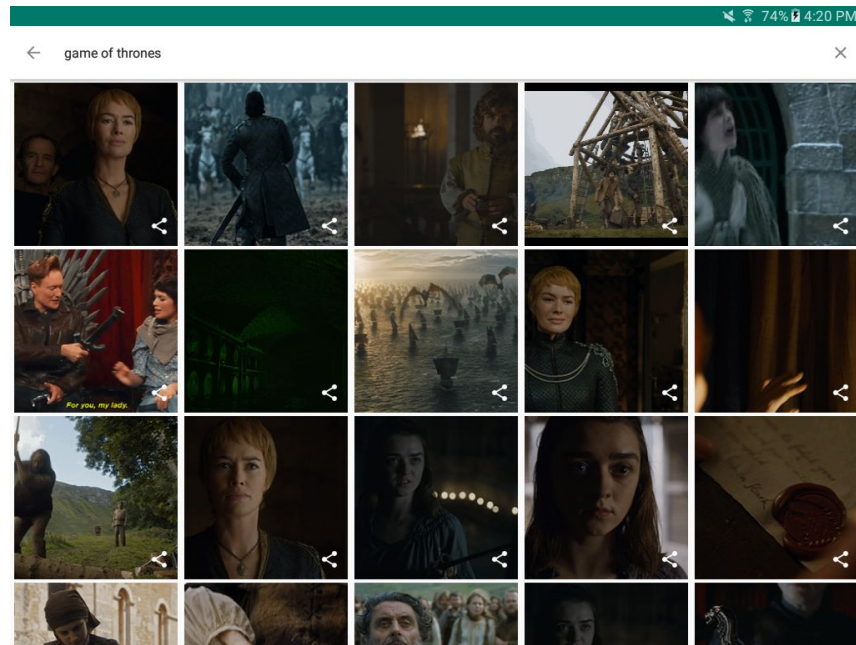
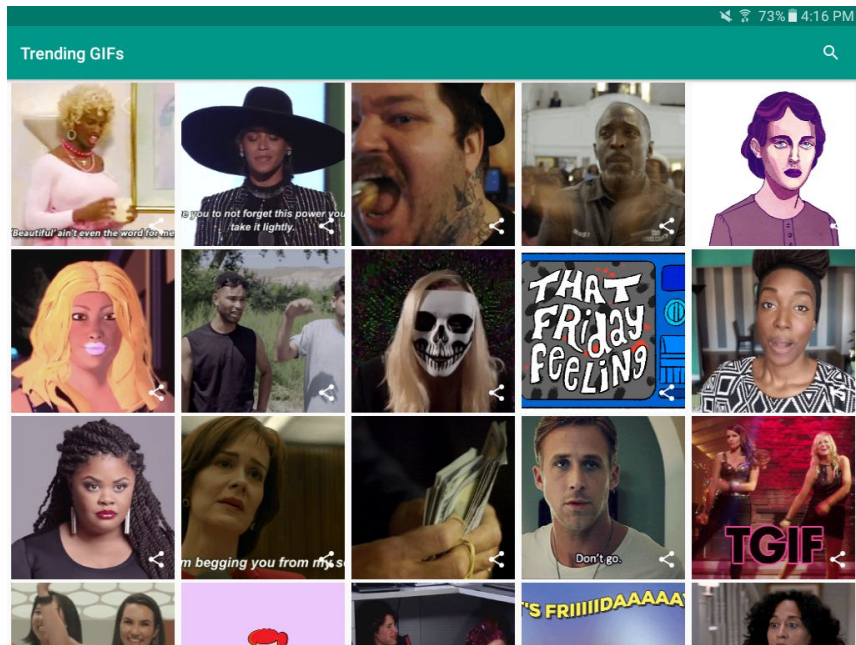


**source allies**

# Today's Workshop

- Make an app to find and share GIFs
- Consumes the Giphy API
- Focus on the Android side
  - Android UI/UX
  - No database for this app
  - Backend is already established through Giphy
  - I have made a Giphy API client to wrap the network calls

# Today's Workshop



# Getting Started

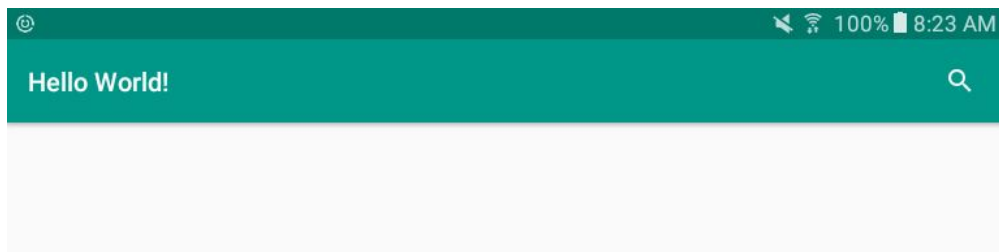
- GitHub Project: <https://github.com/klinker24/giphy-demo>
  - Checkout the “part\_1” branch to get started
  - There are new branches for every step, but feel free to continue off of your old code instead as you finish the previous parts.
- In Android Studio hit the *Open Existing Android Studio Project* button
- Open the root directory for *giphy-demo*

# Some Notes

- Giphy API: <https://github.com/Giphy/GiphyAPI>
- *AndroidManifest.xml*
- Resources for different configurations
- Activities, Services, and Receivers
- Gradle Build System: <http://goo.gl/vJQGVq>

# Part 1: Hello World in App Bar

1. Look at the *TrendsActivity*. This is the “Launcher Activity” that is started when we open the app
2. Look at *TrendsActivity* inheritance
3. *SearchActivity* manages our *Toolbar*. How do we change the text?



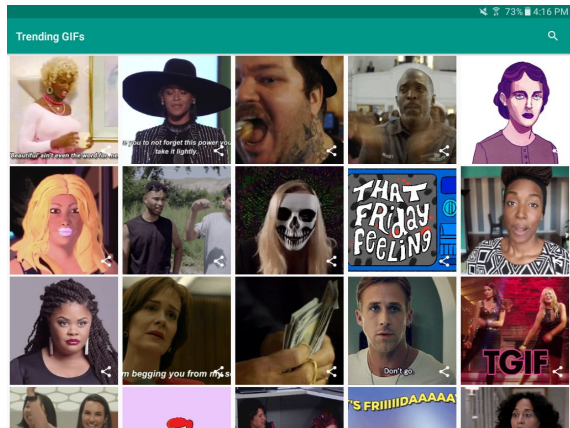
# Part 2: Editing the layout

1. *GiphyActivities* uses the *activity\_main.xml* layout
2. Add a *RecyclerView* to it
3. Set up that *RecyclerView* within the *GiphyActivity*
4. Change the text in the app bar to “Trending GIFs”

```
<android.support.v7.widget.RecyclerView  
    android:id="@+id/recycler_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_marginLeft="3dp"  
    app:layout_behavior=  
        "@string/appbar_scrolling_view_behavior" />
```



# Part 3: Implement the Adapter



1. Why didn't the *RecyclerView* show anything?
2. Adapters control and populate lists in *Android* (*ListView/RecyclerView*)
3. Why is it called a *RecyclerView*?
4. Look at the *adapter\_item\_gif.xml*
5. Implement adapter and Glide to load data
6. Where do these libraries come from?

# Part 4: A simple AsyncTask

1. *AsyncTask*'s control background threads in *Android*
2. We were already using one, but now we are going to implement it ourselves

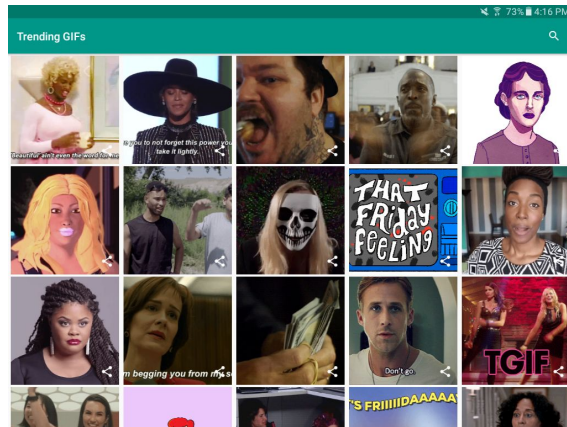
```
@Override  
protected List<Gif> doInBackground(Void... voids) {  
    return null;  
}
```

```
@Override  
protected void onPostExecute(List<Gif> gifs) {  
}
```

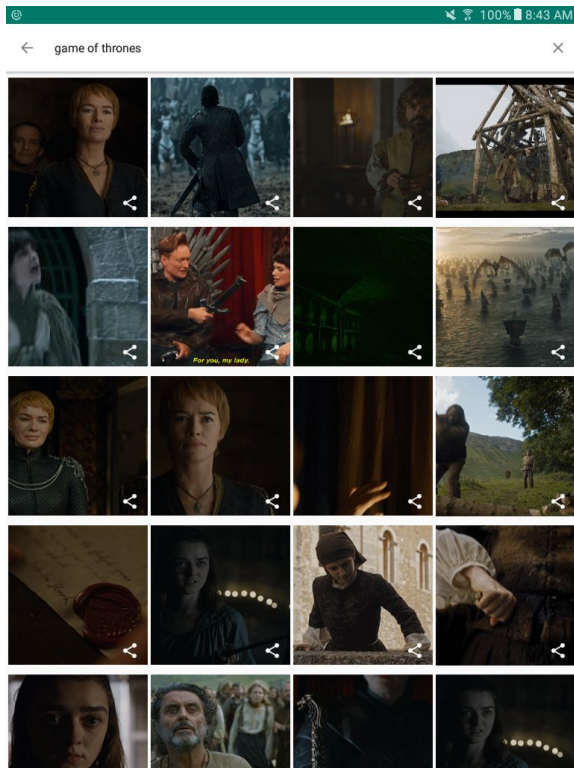
3. Add a *doInBackground()* and *onPostExecute()* to *GiphyTask*

# Part 5: Downloading and Sharing GIFs

1. Go back to the *adapter\_item.xml* and add a share button
2. Check out the more complex *ShareGifTask*
3. Use it in the *GifViewHolder*



# Part 6: Implementing Searches



1. Look at *SearchActivity* to see how the *SearchView* works
2. Static factory methods are a great way to send arguments with an *Intent*
3. How do we use those extras on the *Intent*?

# Other Topics to Explore

- Databases
- Custom Widgets
  - Ex: *SimpleVideoView* or *LabeledEditText*
- *IntentServices*
- *BroadcastReceivers*
- Material Design