



# Exercise Session 4

30.09.2018  
Lists & Adapters

# Are you set up?

In case you need any help, don't be shy, there's plenty of mentors around you!

For references, feel free to look in the presentation slides!

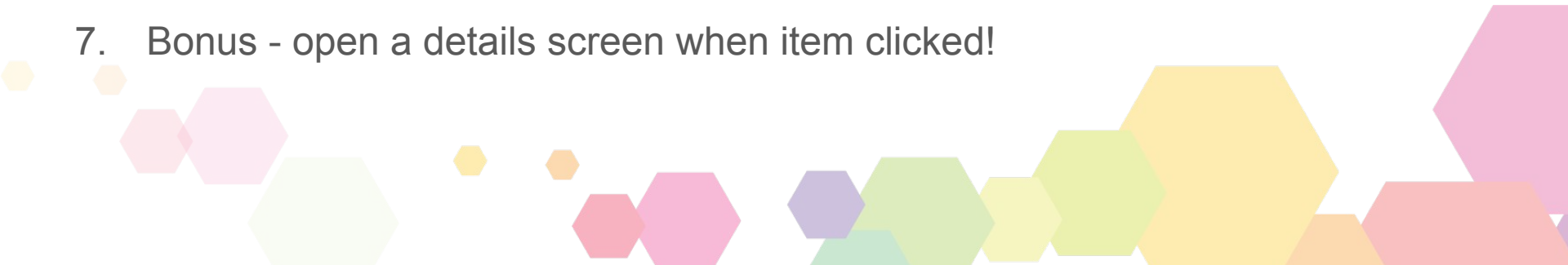
Also, feel free to look at the git project Exercise #4 branch of the lecture from this link on GitHub:

<https://github.com/android-academy-minsk/fundamentals-2019-exercise/tree/ex-4-lists-and-adapters>



# Overview

1. We will create a new Activity (MoviesActivity).
2. You will get the data that you need to show (local data for now).
3. Create XML files with your desired design (activity layout and list item layout).
4. Create a RecyclerView and set its LayoutManager.
5. Create a custom RecyclerView.Adapter
6. Set the adapter to your RecyclerView
7. Bonus - open a details screen when item clicked!



# Step 1 - Create a new Activity!

Create a new Activity (MoviesActivity) in Android Studio

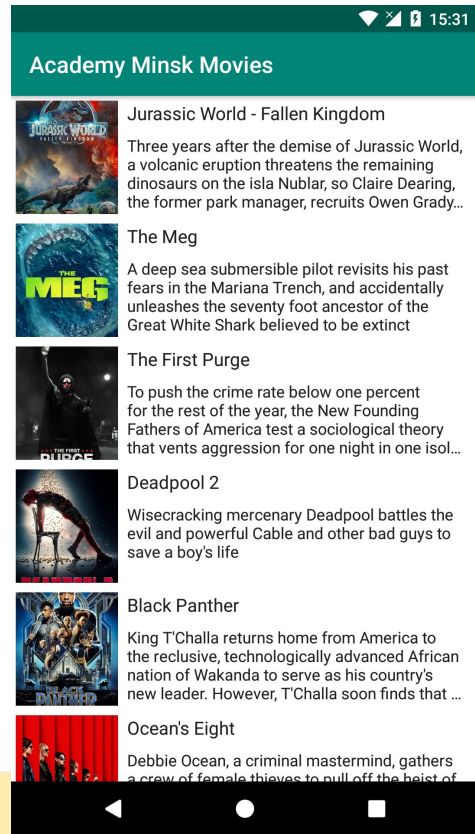


# Step 2 - Add our data!

we're aiming to this end result...

So you'll need movies title, poster, overview etc.

(You can also create your own if you wish)



# Step 3 - Create your XML file

Our row has three objects: name, image and overview.

Create an XML file for our row.

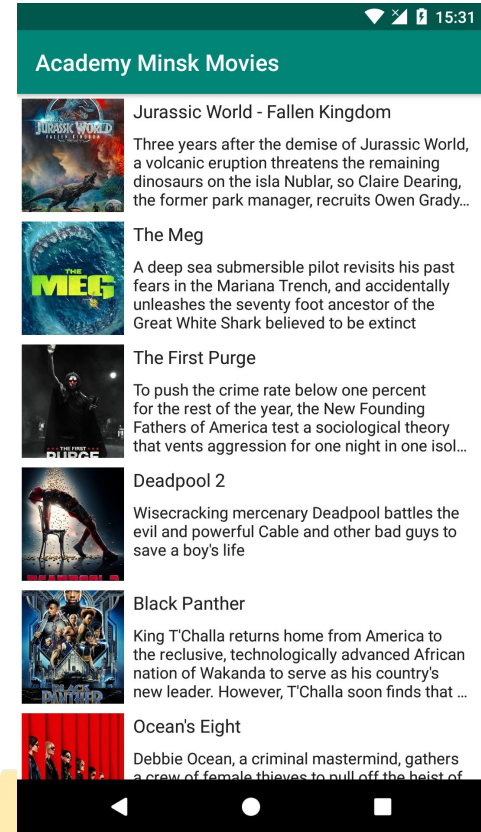
You may use any layout you want:

Constraint, Relative, Linear etc...

Place your text views inside the layout relative to the image,  
customize them the way you see fit  
(textSize, textColor, textStyle and so on)

Bonus

Use the resources files in our xml to store the values you're using!



## Step 4 - Create a RecyclerView

Create your RecyclerView.

\*\* Give your RecyclerView an id so you can access it from your Activity code.



# Step 5 - Create a custom RecyclerView.Adapter

Create a new class and give it your custom name.

Extend `RecyclerView.Adapter<ViewHolder>` and implement the required methods.

Create a class that extend `RecyclerView.ViewHolder` and use it as the type of the generic Adapter (`RecyclerView.Adapter<YOUR_CLASS>`)





## Step 6 - Set the adapter to the RecyclerView

In your code, simply set the Adapter to your RecyclerView.



## Step 7 (Bonus) - Add a click listener

1. Add a callback property to your RecyclerView.Adapter that provides an Int as position.
2. In your RecyclerView.ViewHolder you get an instance of a view, use it to set callback.
3. In your onClick method call the relevant callback of you interface.
4. Use getAdapterPosition() function of RecyclerView.ViewHolder to get the position of the clicked item in the list.



## Step 8 (Bonus) - Open Movie details screen

1. Add helper method to DetailsActivity that creates an Intent.
2. Call startActivity() with an Intent for a specific movie
3. Update DetailsActivity onCreate() method to get a movie from Intent and update views with the actual data.
4. You may want to update its layout as well to use @tools.



# Summary

A ListView is a good tool for simple things such as showing simple views in an AlertDialog, Spinner and such widgets.

For more complex views it is recommended to use a RecyclerView!

What you did in this exercise is just the tip of the power of The RecyclerView!



# Resources

## RecyclerView docs:

<https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView>

<https://developer.android.com/guide/topics/ui/layout/recyclerview>

## RecyclerView.Adapter docs:

<https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView.Adapter.html>

## RecyclerView.ViewHolder docs:

<https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView.ViewHolder.html>

## Movies images:

[Jurassic World](#), [The Meg](#), [The First Purge](#), [Deadpool 2](#), [Black Panther](#), [Ocean's Eight](#), [Interstellar](#), [Thor - Ragnarok](#), [Guardians of the Galaxy](#).

**Kudos!!**

You're all done with exercise 4!

See you next time!

