

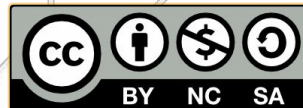
# Android Development

## Adapters and Recycler View

**SoftUni Team**  
Teodor Kostadinov



**SoftUni  
Foundation**



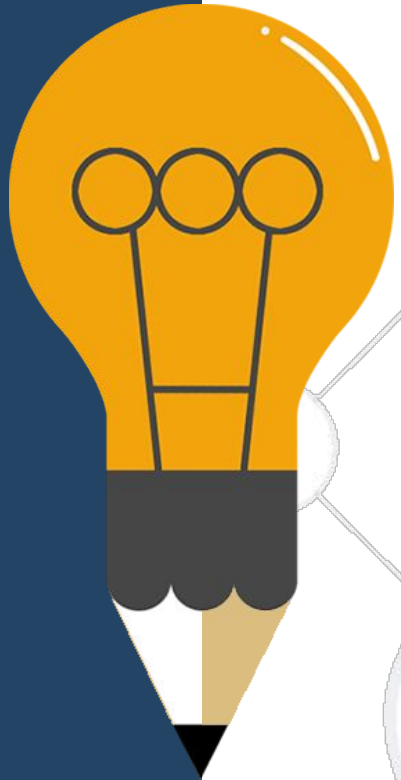
**Software University**

<http://softuni.bg>

A background network diagram consisting of several white circles of varying sizes connected by thin grey lines. The circles are arranged in a non-uniform, interconnected pattern across the slide.

**sli.do**  
**#Android**

# Just a second!



Do you remember what we talked about last time?

Please fill this form:

<https://forms.gle/ArDYAQPSquoqWdJGA>

This will help me understand what you remember from our previous lecture.

# But first - Toolbar!

- ActionBar was added in API level 11, in 21 was replaced with Toolbar
- ActionBar was hard to customize, was editable only in java code, was not part of the view hierarchy
- Toolbar solves those problems.
  - It's just a View
  - It can be anywhere in the hierarchy
  - Can be assigned as an ActionBar



Source: <https://developer.android.com/reference/android/app/ActionBar>  
<https://developer.android.com/reference/android/support/v7/app/ActionBar>  
<https://developer.android.com/reference/android/widget/Toolbar>

- Add implementation 'com.android.support:appcompat-v7:21.0.+' in the build.gradle file
- To set a Toolbar as an ActionBar, use `setSupportActionBar(toolbar);`
- To avoid action bar clashes, use the theme `Theme.AppCompat.Light.NoActionBar`
  - change it in `res/values/styles`

- Buttons located in the ActionBar are called Action Buttons
- They should be really actions, not navigation - search, send..
- If more actions than space allows, Overflow Menu is shown, which shows a dropdown of all hidden actions
- Actions are defined in xml in menu folder

- ActionBar's Logo can be replaced with an Up arrow
- Up should take the user to the previous activity in the hierarchy
- There should be a difference between Up and Back
- Overriding `onOptionsItemSelected` allows you to detect a click on the Up and actions
- Overriding `onBackPressed` allows you to detect when back is pressed



# **Demo**

## **ActionBar vs ToolBar**



- Before: findViewById
- Now:

```
ActivityLoginBinding binding = DataBindingUtil.setContentView(this,  
R.layout.activity_login);  
binding.btnLogin.setText(...);
```

```
binding.btnLogin.setText(...)
```

- To enable it change the gradle file

```
dataBinding {  
    enabled = true  
}
```

- Layouts should have a parent tag `<layout>`
- This allows the system to automatically generate an object, which has all view bindings



# **Demo**

## **View Binding**

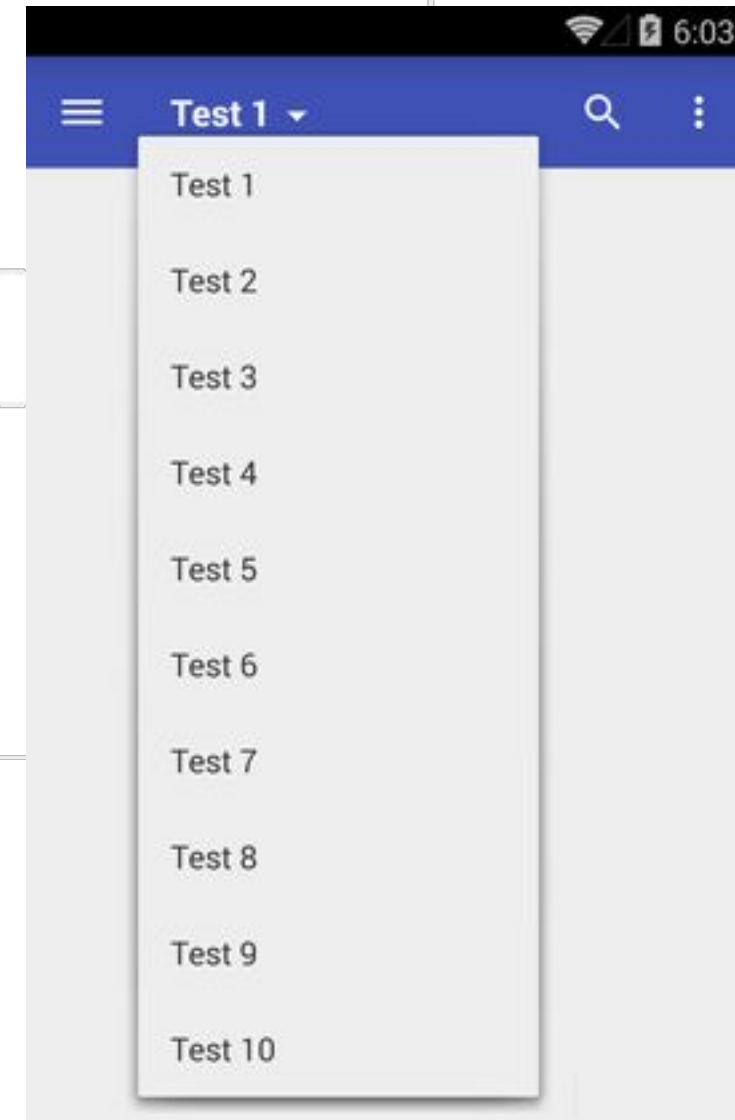
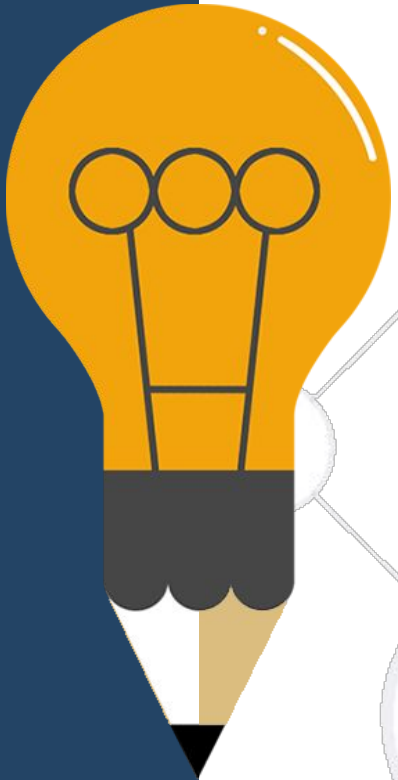
- View, which shows its elements in a dropdown list

```
// Create an ArrayAdapter, add items and a default spinner layout
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
    items, android.R.layout.simple_spinner_item);
// Specify the layout to use when the list of choices appears
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
// Apply the adapter to the spinner
spinner.setAdapter(adapter);
```

# Let's do this!

Write the following app:  
Create a toolbar. Have an icon,  
action button and overflow icon.

Instead of a title, have a spinner.



- Adapters connect the View with the data
  - An adapter has:
  - Method for item count
  - Method for getting an item
  - Method for getting a view with data set from the data of an item
  -
- 
- We used to use different adapters
  - ArrayAdapter - for data in arrays
  - CursorAdapter - for data in the database
  - BaseAdapter - for general use
  - Now we don't use them anymore.

- Views that require an Adapter to work
- ListView - shows items in a list
- GridView - in a grid
- Gallery - shows items screen by screen
- These views are obsolete and probably removed.

- Displays a collection of data

Java:

```
mRecyclerView = (RecyclerView) findViewById(R.id.my_recycler_view);  
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));  
mAdapter = new MyAdapter(myDataset);  
mRecyclerView.setAdapter(mAdapter);
```

XML:

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



- Should inherit `RecyclerView.Adapter<MyAdapter.MyViewHolder>`
- Should override `onCreateViewHolder`
  - used to inflate the View and create the ViewHolder object
- Should override `onBindViewHolder`
  - used to set values to the view inside the ViewHolder

- Avoids view binding (`findViewById`) every time the view is scrolled
- Simple class that has fields - the Views in one item of the list
- It works like this:
  - When a new View is created, it's views are passed to a view holder to keep them safe
  - When a list item should be shown, the view holder for that item is used, it's views are changed according to the data



# **Demo RecyclerView**

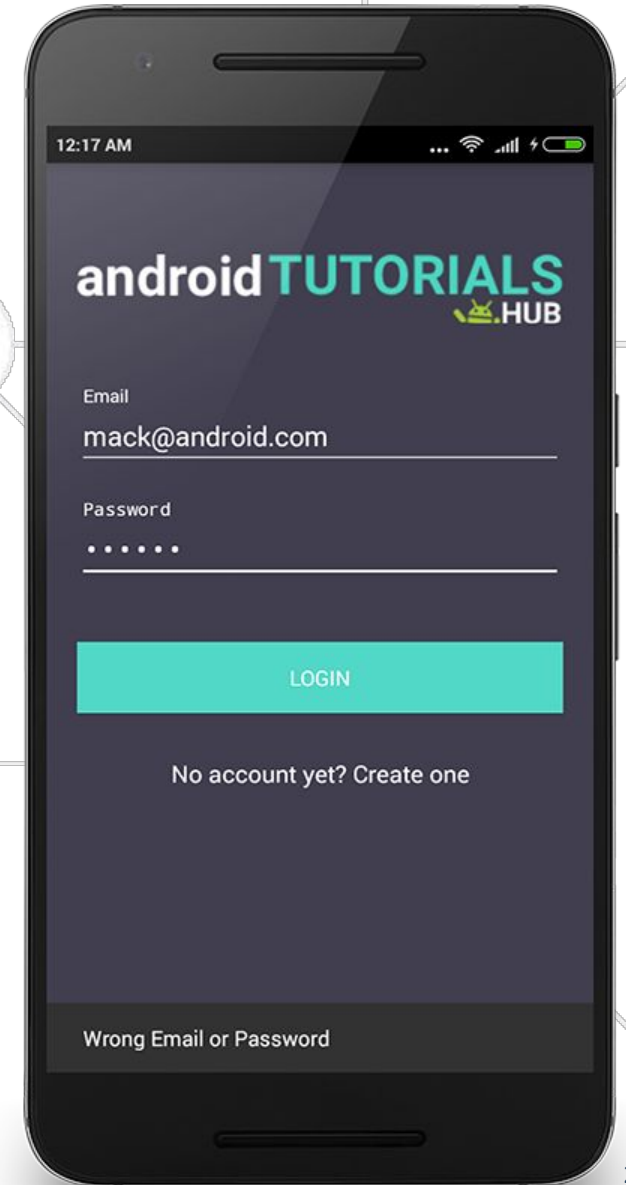
# Let's do this!

Add to your app:

Below Login button add create account text. Pressing the text takes you to another screen.

RegisterScreen has:

- email
- password, repeat password
- submit button that sends the data via email
- Contact us button that dials a number
- Has a check for wrong email format and password mismatch



**There are a lot of Views. One of the important ones is RecyclerView.**

Resources:



# Homework (1)

Create the 9gag app.

It should have a vertical list of items. Each item has:

- title
- image
- points and comments
- vote up and down
- comment
- share

Pressing up button would disable the button. Pressing down will un-press up and make down button pressed. Both buttons cannot be pressed at the same time. Pressing the comment button would take the user to another screen. Pressing Share would share the title and image with Facebook, Twitter or other installed apps.

Populate at least 10 different items in the list.



14,565 Points · 149 Comments



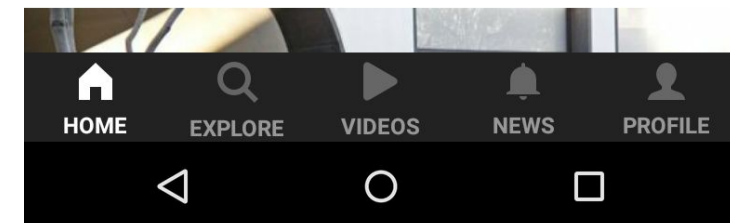
**It's a little flaw, but it drives me crazy sometimes.**



9,845 Points · 196 Comments



**A 1992 Jaguar xj220.**





# Homework (2)

Upgrade the 9gag app.

When the user presses the comment button, it opens a new screen.

The screen shows all the comments from the users, datetime sent and username.

User can enter and add a new comment. It appears at the bottom of the screen.

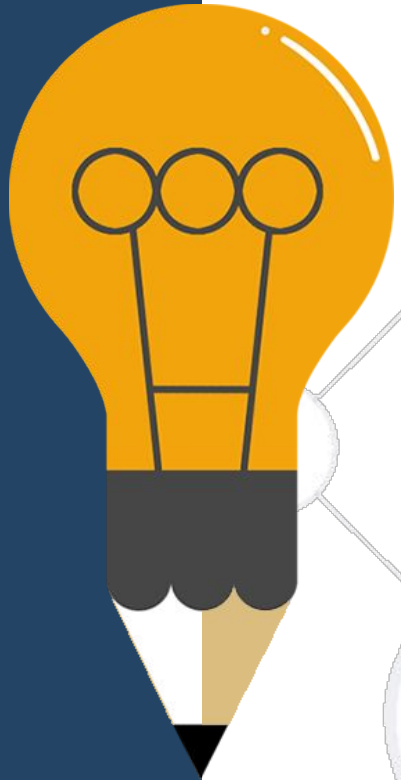
Data should not be remembered. Exiting the screen will clear all the entered comments.



# Homework (3)

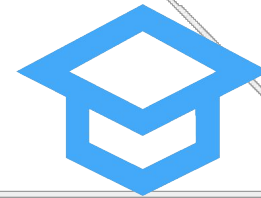
Upgrade the 9gag app.

- Add a new type of item in the list - Ad
- Ads have only an image and a button (Buy)
- Pressing the Buy button opens Google in a browser

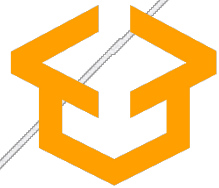




# Questions?



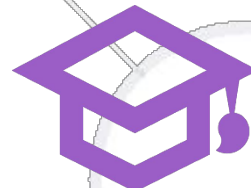
**SoftUni**



**Software  
University**



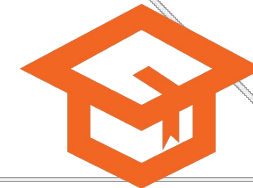
**SoftUni  
Svetlina**



**SoftUni  
Creative**



**SoftUni  
Digital**



**SoftUni  
Foundation**



**SoftUni  
Kids**



# СофтУни диамантени партньори

INDEAVR

Serving the high achievers



INFRAGISTICS®



**SoftwareGroup**  
*doing it right*



**XS**software

NETPEAK

**SUPER  
HOSTING  
®.BG**

# СофтУни диамантени партньори



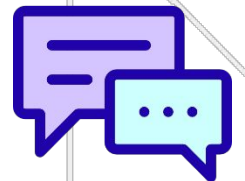
**LIEBHERR**



# Trainings @ Software University (SoftUni)



- Software University – High-Quality Education, Profession and Job for Software Developers
  - [softuni.bg](http://softuni.bg)
- Software University Foundation
  - <http://softuni.foundation/>
- Software University @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)
- Software University Forums
  - [forum.softuni.bg](http://forum.softuni.bg)



# Tabs and Bottom Sheet

- Tabs need external library
  - compile 'com.android.support:design:23.0.1'
- Each tab should be a new fragment
- Change between fragments happens with a ViewPager
- ViewPager uses an Adapter