

CS 213 – Software Methodology

Spring 2017

Sesh Venugopal

Lecture 22 – Apr 13
Android Programming

ActionBar/Up Navigation

Movies:

Action/App Bar and Up Navigation with Backward Compatibility

Action/App Bar Replaces Menu Bar of old

The Action Bar is a dedicated area that appears at the top of every screen of an application

It offers consistent way to navigate around the application, without interference with other apps. For instance, you use the “Up” navigation arrow to go back to the previous screen in your current navigation flow, rather than the back facility in the phone’s set up.

The Action Bar is where you want to put global as well as contextual menu options, including the ability to use the overflow area.

See <https://developer.android.com/design/patterns/actionbar.html>

Adding the App Bar

See <https://developer.android.com/training/appbar/index.html>

The native Action Bar for different versions of Android (starting with API version 11, or Android 3.0) offer different capabilities.

This makes the UI for an app vary depending on user's Android setup, which is not good.

The Android folks recommend that instead of the native Action Bar, you use the **ToolBar** widget instead, which is maintained in a consistent manner by the **v7 compat** support library so you can run the same app on different platforms with a consistent action bar experience.

Ensuring Backward Compatibility

To use the support library, make sure you install the Android Support Repository and Android Support Library under Extras (last set of installs) in the SDK manager

And code all your activity classes that have a UI presence to extend `AppCompatActivity`

```
import android.support.v7.app.AppCompatActivity;  
...  
public class Routes extends AppCompatActivity {
```

(The Backward Compatibility option is automatically checked when using the project set up wizard, producing the above boiler-plate code)

Steps to add the Toolbar

Step 1:

Add the layout for the Toolbar as the top item in each screen's layout file

```
<android.support.v7.widget.Toolbar
    android:id="@+id/my_toolbar"
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:background="?attr/colorPrimary"
    android:elevation="4dp"
    android:theme="@style/ThemeOverlay.AppCompat.ActionBar"/>
```

Add this to the `movies_list.xml` and `add_edit_movie.xml` files

Step 2:

In the manifest file, change the theme to `NoActionBar` so that the native Action Bar is not deployed:

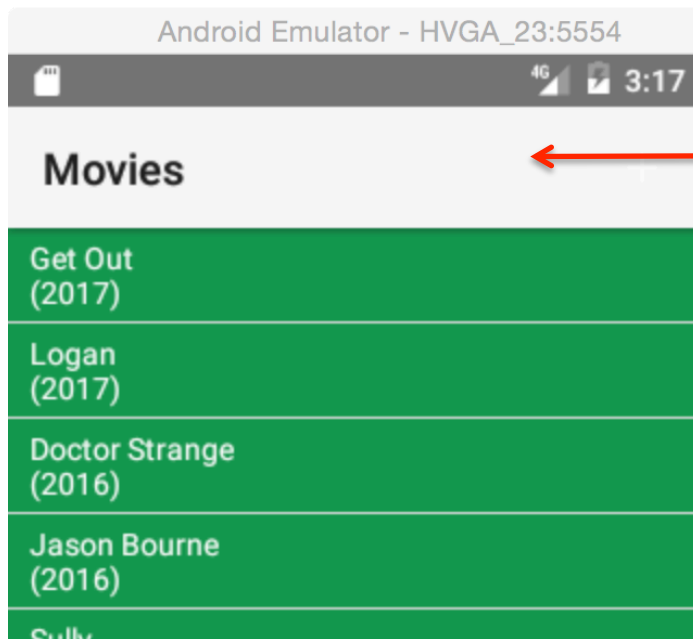
```
<application
    android:theme="@style/Theme.AppCompat.Light.NoActionBar"
```

Steps to add the Toolbar

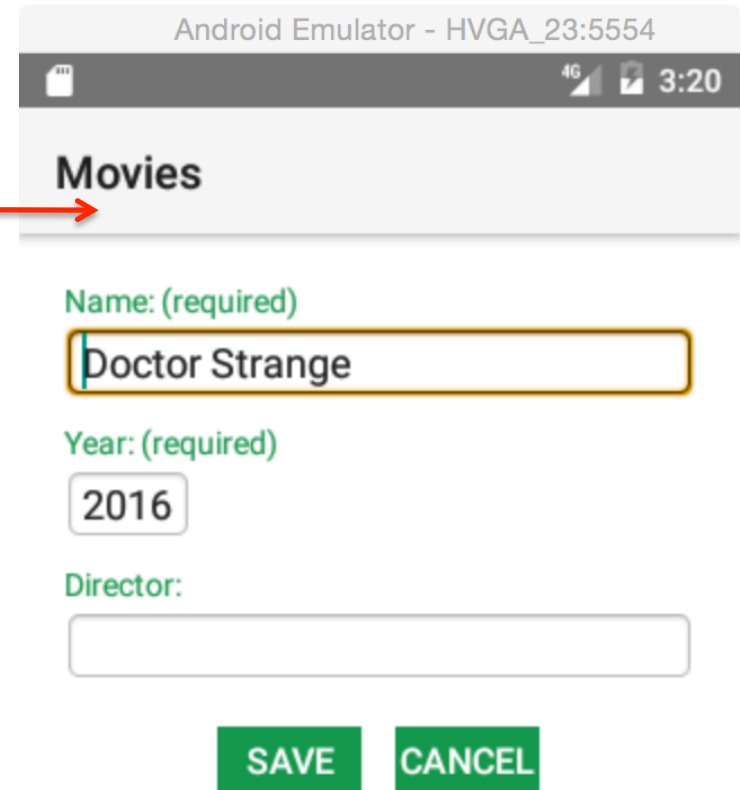
Step 3:

In each activity's `onCreate` method, get the `ToolBar` widget, and add it as follows, so that the `ToolBar` functions as the action bar

```
ToolBar myToolbar = (ToolBar) findViewById(R.id.my_toolbar);  
setSupportActionBar(myToolbar);
```



Toolbar
(standing in
for Action Bar)



Adding Up Navigation to Add/Edit Movie Screen

Step 4:

In the `AddEditMovie` class's `onCreate` method, do this:

```
ActionBar ab = getSupportActionBar();  
ab.setDisplayHomeAsUpEnabled(true);
```

true means go to
parent, not top

In the manifest file, for the `AddEditMovie` activity, do this:

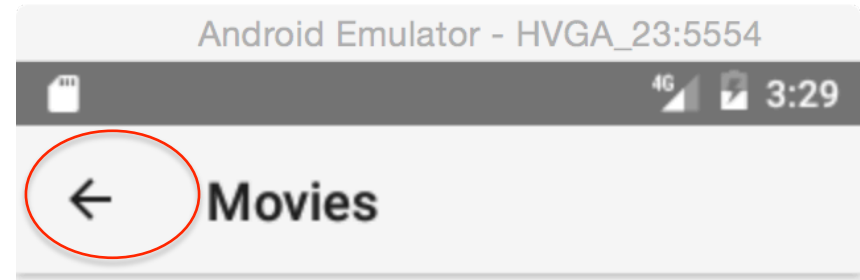
```
...  
<activity android:name=".AddEditMovie"  
    android:parentActivityName=".Movies">  
    <meta-data
```

For Android 4.1 and
higher

For Android 4.0 and lower, with support library

```
        android:name="android.support.PARENT_ACTIVITY"  
        android:value=".Movies" />  
    </activity>
```

See <https://developer.android.com/training/implementing-navigation/ancestral.html>



Name: (required)

The Martian

Year: (required)