

Displaying Data From Code

This lesson explains how to set all the data from code and write the logic for the image with back icon.

WE'LL COVER THE FOLLOWING ^

- Tools attributes
- Activity declaration
- Launch details screen
- Setting data from code
- Image with the back icon

Tools attributes

In the previous lesson, we defined test layout data inside the XML file, but this data will be dynamically set from code. If we leave things as they are right now, we will see test data before real data is loaded.

To avoid this situation, we can remove the test data, but this means that it will be hard to understand what data should be displayed without digging into code. There is a better way to handle this situation - tools attributes.

Tools attributes live in a special namespace

`xmlns:tools="http://schemas.android.com/tools"`. It's a set of attributes to enable design-time features. These attributes are automatically removed during the compilation process.

First, let's add a tools namespace to the root layout.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```



```
...  
</androidx.constraintlayout.widget.ConstraintLayout
```

activity_blog_details.xml

Next, replace all `android:text`, `android:src` and `android:rating` with `tools:text`, `tools:src` and `tools:rating`. This will allow data to be set dynamically from the code.

Activity declaration

Let's create *BlogDetailsActivity* class and bind our layout to this activity by using the `setContentView` method.

```
public class BlogDetailsActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_blog_details);  
    }  
}
```

Register *BlogDetailsActivity* in the *AndroidManifest.xml* file.

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.travelblog">  
    <application  
        android:theme="@style/AppTheme"  
        android:label="Travel Blog">  
        <activity android:name=".LoginActivity">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
        <activity android:name=".MainActivity" />  
        <activity android:name=".BlogDetailsActivity" />  
    </application>  
</manifest>
```

AndroidManifest.xml

Launch details screen

Right now we launch *MainActivity* after *LoginActivity*. Let's add the code to temporarily launch *BlogDetailsActivity* immediately when *MainActivity* is opened.



```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        TextView mainTextView = findViewById(R.id.mainTextView);
        mainTextView.setText("Hello educative.io");

        startActivity(new Intent(this, BlogDetailsActivity.class));
    }
}
```

Setting data from code

Now let's bind the view from XML to Java objects via the `findViewById` method and set the data from code.

To display the images, we are going to use the `setImageResource` method with ids of the images as a parameter. To show the stars, we are going to use the `setRating` method which accepts `float` value as a parameter.



```
public class BlogDetailsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_blog_details);

        ImageView imageMain = findViewById(R.id.imageMain);
        imageMain.setImageResource(R.drawable.sydney_image);

        ImageView imageAvatar = findViewById(R.id.imageAvatar);
        imageAvatar.setImageResource(R.drawable.avatar);

        TextView textTitle = findViewById(R.id.textTitle);
        textTitle.setText("G'day from Sydney");

        TextView textDate = findViewById(R.id.textDate);
        textDate.setText("August 2, 2019");

        TextView textAuthor = findViewById(R.id.textAuthor);
        textAuthor.setText("Grayson Wells");

        TextView textRating = findViewById(R.id.textRating);
        textRating.setText("4.4");

        TextView textViews = findViewById(R.id.textViews);
        textViews.setText("(2687 views)");

        TextView textDescription = findViewById(R.id.textDescription);
```

```

        textDescription.setText("Australia is one of the most popular travel destinations in

        RatingBar ratingBar = findViewById(R.id.ratingBar);

        ratingBar.setRating(4.4f);
    }
}

```

Image with the back icon

The last final step is to close the screen when we click on the image with the back icon. We can do this by simply adding a click listener to the image view and calling the `finish()` method inside.

```

public class BlogDetailsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_blog_details);
        ...
        ImageView imageBack = findViewById(R.id.imageBack);
        imageBack.setOnClickListener(v -> finish());
    }
}

```

Hit the *run* button to try it yourself.

```

package com.travelblog;

import android.os.Bundle;
import android.widget.ImageView;
import android.widget.RatingBar;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class BlogDetailsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activitiy_blog_details);

        ImageView imageMain = findViewById(R.id.imageMain);
        imageMain.setImageResource(R.drawable.sydney_image);

        ImageView imageAvatar = findViewById(R.id.imageAvatar);
        imageAvatar.setImageResource(R.drawable.avatar);

        ImageView imageBack = findViewById(R.id.imageBack);
        imageBack.setOnClickListener(v -> finish());

        TextView textTitle = findViewById(R.id.textTitle);
    }
}

```

```
textTitle.setText("G'day from Sydney");

TextView textDate = findViewById(R.id.textDate);
textDate.setText("August 2, 2019");

TextView textAuthor = findViewById(R.id.textAuthor);
textAuthor.setText("Grayson Wells");

TextView textRating = findViewById(R.id.textRating);
textRating.setText("4.4");

TextView textViews = findViewById(R.id.textViews);
textViews.setText("(2687 views)");

TextView textDescription = findViewById(R.id.textDescription);
textDescription.setText("Australia is one of the most popular travel destinations in

RatingBar ratingBar = findViewById(R.id.ratingBar);
ratingBar.setRating(4.4f);
    }
}
```

In the next lesson, we will learn how to load images from the Internet.