

Android Splash Screen

Android Splash Screen is usually shown at the start of the application for 2 or 3 seconds of time.

Main purpose of **Android Splash Screen** is to show logo and name of the application to the user.

We will create three types of splash screens in this article.

Sometimes it is also used for fetching required data from remote server before starting the app.

Splash screen is a simple screen or activity used to display some progress when application is launched.

Some developers use splash screen to just show company logo or to engage user while some important data is loaded in background.

After few seconds or minutes, splash screen is stopped and any other activity is started.

An *Android* app takes some time to start up, especially when the app is first launched on a device.

A splash screen may display start up progress to the user or to indicate branding.

Latest purpose of splash screen can be found as you can ask for all permissions you are using in application.

In all android versions after MARSHMALLOW, you have to request for permission to user.

1. Android Simple Splash Screen Example

You can implement Splash Screen in many ways in Android. But I have shown the easiest and quickest way to implement a splash screen.

Following is the output of splash demo:

Step 1. Create a new project

Create a new project in Android Studio with Empty Activity.

Step 2. Code for splash

Copy and paste the following code in **activity_main.xml** file:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="fitXY"
        android:src="@drawable/splash"/>
```

</RelativeLayout>

We have taken an ImageView with full size of screen.

You need to make a separate image for splash.

This image may contain company name and/or logo with or without brand's slogan.

Splash Image Size

You need to make different size of splash images for different android device screens.

There may be a small variations among various image sizes and screen sizes.

I am going to give you a dimensions which can be used generally in most of the cases.

- LDPI:
 - Portrait: **200x320px**
 - Landscape: **320x200px**
- MDPI:
 - Portrait: **320x480px**
 - Landscape: **480x320px**
- HDPI:
 - Portrait: **480x800px**
 - Landscape: **800x480px**
- XHDPI:
 - Portrait: **720px1280px**
 - Landscape: **1280x720px**
- XXHDPI:
 - Portrait: **960px1600px**
 - Landscape: **1600x960px**
- XXXHDPI:
 - Portrait: **1280px1920px**
 - Landscape: **1920x1280px**

Use images of above size for making your splash screen responsive among all the mobile and tablet devices.

Copy and paste following code in **MainActivity.java**

```
import android.content.Intent;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    /** Duration of wait */
    private final int SPLASH_DISPLAY_LENGTH = 2000; //splash screen will be shown for 2
seconds
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
```

```

        Intent mainIntent = new Intent(MainActivity.this, WelcomeActivity.class);
        startActivity(mainIntent);
        finish();
    }
}, SPLASH_DISPLAY_LENGTH);
}
}

```

Introduction of HANDLER Class

The handler class lets you process and send message to other threads, which contain data, and runnable objects that belonging to a thread.

A handler will let you execute message objects and runnable objects at a specified time in the future.

Each instance of a handler will belong to a single thread only when declared.

Runnable objects are those objects that contain commands that will be executed to obtain results in your program.

To explain it to you plainly, a handler be used to execute a thread.

It will also allow various threads that make up a program to communicate with each other to prevent conflicts.

Explanation of MainActivity

We have used Handler class of **JAVA** to implement logic of Splash Screen.

A handler has a method called **run()**.

In this **run()** method, you can write your required source code or give some instruction to your app.

Here in this example, I have tell my app to open **WelcomeActivity**.

Now with the help of Handler, you can delayed the execution of the code written in **run()** method.

You can give time duration in mili seconds to delay the execution of **run()** method.

In above example, I have taken 2000 mili seconds as delay.

I have used integer variable for this (**SPLASH_DISPLAY_LENGTH**).

Right now value of **SPLASH_DISPLAY_LENGTH** is 2000 milliseconds, which is equals to 2 seconds.

You can change this time duration as per your requirements. (Increase this value to shown splash for more time otherwise decrease it.)

finish() method is written after starting a new activity.

Splash screen is shown only at the opening of the app and when the user presses the back button to exit the app, splash screen should not be shown at that time.

For this purpose, **finish()** method will help us. It will remove the splash screen activity from the backstack so when the user presses back button, he will directly left the app.

Second Method

You can implement the splash screen with Thread class also.

Sample source code for using Thread is as the below source code

```
Thread splashTread = new Thread() {
    @Override
    public void run() {
        try {
            int waited = 0;
            while (_active && (waited < SPLASH_DISPLAY_LENGTH)) {
                sleep(100);
                if (_active) {
                    waited += 100;
                }
            }
        } catch (Exception e) {}
        finally {
            startActivity(new Intent(SplashScreen.this,
                                    MainActivity.class));
            finish();
        }
    };
};
splashTread.start();
}
```

we are implementing a **while()** method in a thread.

I have declared one integer variable called **waited**. Value of **waited** is zero at the starting.

Value of **waited** is increased by 100 in the every iteration of the while loop.

when the value of **waited** variable is greater than **SPLASH_DISPLAY_LENGTH** , **finally()** method is called by the compiler.

In the **finally()** method, first activity of the android app is called.

To make splash screen with **thread**, **replace the above code** with the **following code** in our existing example

```
new Handler().postDelayed(new Runnable() {
    @Override
    public void run() {

        Intent mainIntent = new Intent(MainActivity.this, WelcomeActivity.class);
        startActivity(mainIntent);
        finish();
    }
}, SPLASH_DISPLAY_LENGTH);
```

After making these changes you will be able to show the splash screen using thread.