

React Native Splash Screen Setup with Kotlin

Step 1: Create SplashActivity

- Navigate to Android / src / main / java / com / yourpackage / and add a new file named SplashActivity.kt.
- Replace com/example with your actual package name in the code.

SplashActivity.kt

```
package com.yourpackage // Change this to your actual package name
import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import androidx.appcompat.app.AppCompatActivity
import com.yourpackage.R

class SplashActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.splash_screen) // Make sure you have splash_screen.xml in your layout
        directory
        Handler(Looper.getMainLooper()).postDelayed({
            startActivity(Intent(this, MainActivity::class.java))
            finish()
        }, 3000) // Display the splash screen for 3 seconds
    }
}
```

Step 2: Add Splash Screen Image

- Place your splash screen image in the Android / src / main / res / drawable folder.
- The image should be named splash_screen.png.

Step 3: Create Splash Screen Layout

- Inside Android / src / main / res /, create a new folder named layout if it doesn't already exist.
- Add a file named splash_screen.xml in the layout folder with the following content:

//ADD THIS ALL CODE IN SIDE FILE SPLASH_SCREEN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">
    <!-- Customize your splash screen here -->
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/splash_screen" /> <!-- Ensure this refers to your drawable splash screen
image -->
```

</FrameLayout>

Step 4: Add Splash Screen Image to Various Resolutions

- Copy and paste your splash.png image into the following folders for different screen densities:
- mipmap-hdpi
- mipmap-mdpi
- mipmap-xhdpi
- mipmap-xxhdpi
- mipmap-xxxhdpi

Step 5: Update AndroidManifest.xml

- In your AndroidManifest.xml, add the following configuration:

ADD THIS CODE ANDROIDMANIFEST.XML

```
<application
  android:name=".MainApplication"
  android:label="@string/app_name"
  android:icon="@mipmap/ic_launcher"
  android:roundIcon="@mipmap/ic_launcher_round"
  android:allowBackup="false"
  android:theme="@style/AppTheme">
  <!-- SplashActivity as the launcher activity -->
  <activity android:name=".SplashActivity"
    android:exported="true"> <!-- Required for Android 12 and above -->
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
  </activity>
  <!-- MainActivity configuration -->
  <activity
    android:name=".MainActivity"
    android:label="@string/app_name"

    android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|screenSize|smallestScreen
    Size|uiMode"
    android:launchMode="singleTask"
    android:windowSoftInputMode="adjustResize"
    android:exported="true">
  </activity>
</application>
```

Step 6: Install react-native-splash-screen

- Run the following command to install react-native-splash-screen:

```
npm install react-native-splash-screen --save
```

Step 7: Use SplashScreen in Your App Component

- In your main App component (App.js or App.tsx), import SplashScreen from react-native-splash-screen and hide the splash screen after your app is ready:

```
import React, { useEffect } from 'react';
import { View, Text } from 'react-native';
import SplashScreen from 'react-native-splash-screen';
const App = () => {
  useEffect(() => {
    SplashScreen.hide(); // Hide the splash screen
  }, []);
  return (
    <View>
      <Text>Hello world :)</Text>
    </View>
  );
};
export default App;
```