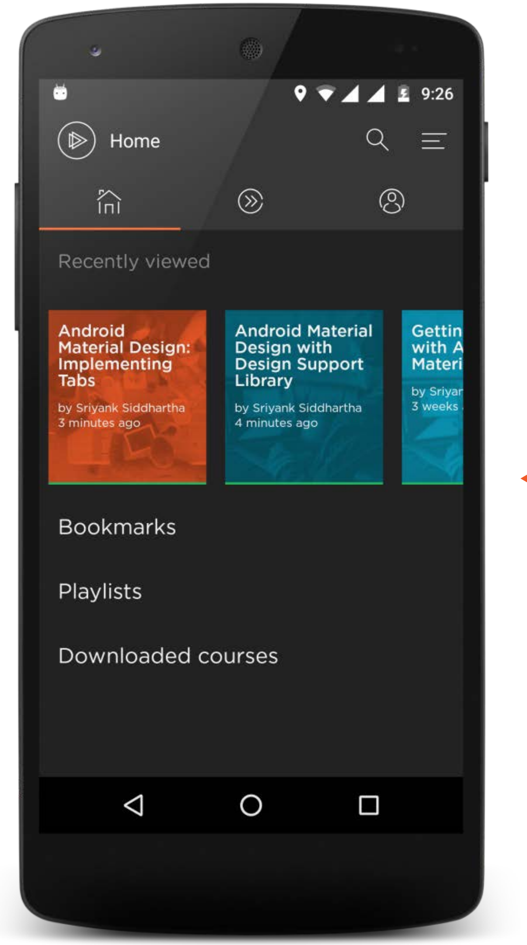
Understanding Activity Lifecycle in Context of Screen Rotation

Sriyank Siddhartha
AUTHOR

linkedin.com/in/sriyank
sriyank.siddhartha@gmail.com

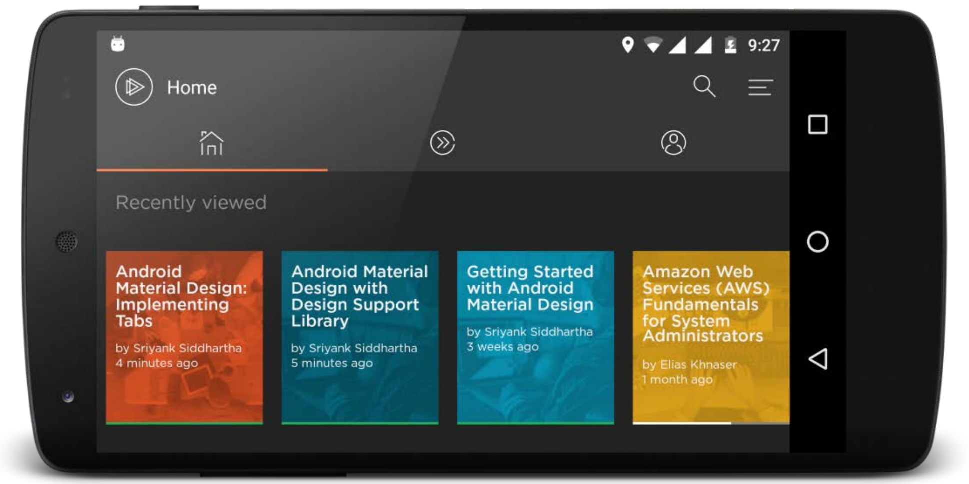
Activity Lifecycle and Screen Orientation

Screen Orientation and Activity Lifecycle



Landscape

Portrait


9:27
Home
Recently viewed
Android Material Design: Implementing Tabs
by Sriyank Siddhartha
4 minutes ago
Android Material Design with Design Support Library
by Sriyank Siddhartha
5 minutes ago
Getting Started with Android Material Design
by Sriyank Siddhartha
3 weeks ago
Amazon Web Services (AWS) Fundamentals for System Administrators
by Elias Khnaser
1 month ago

Conclusion

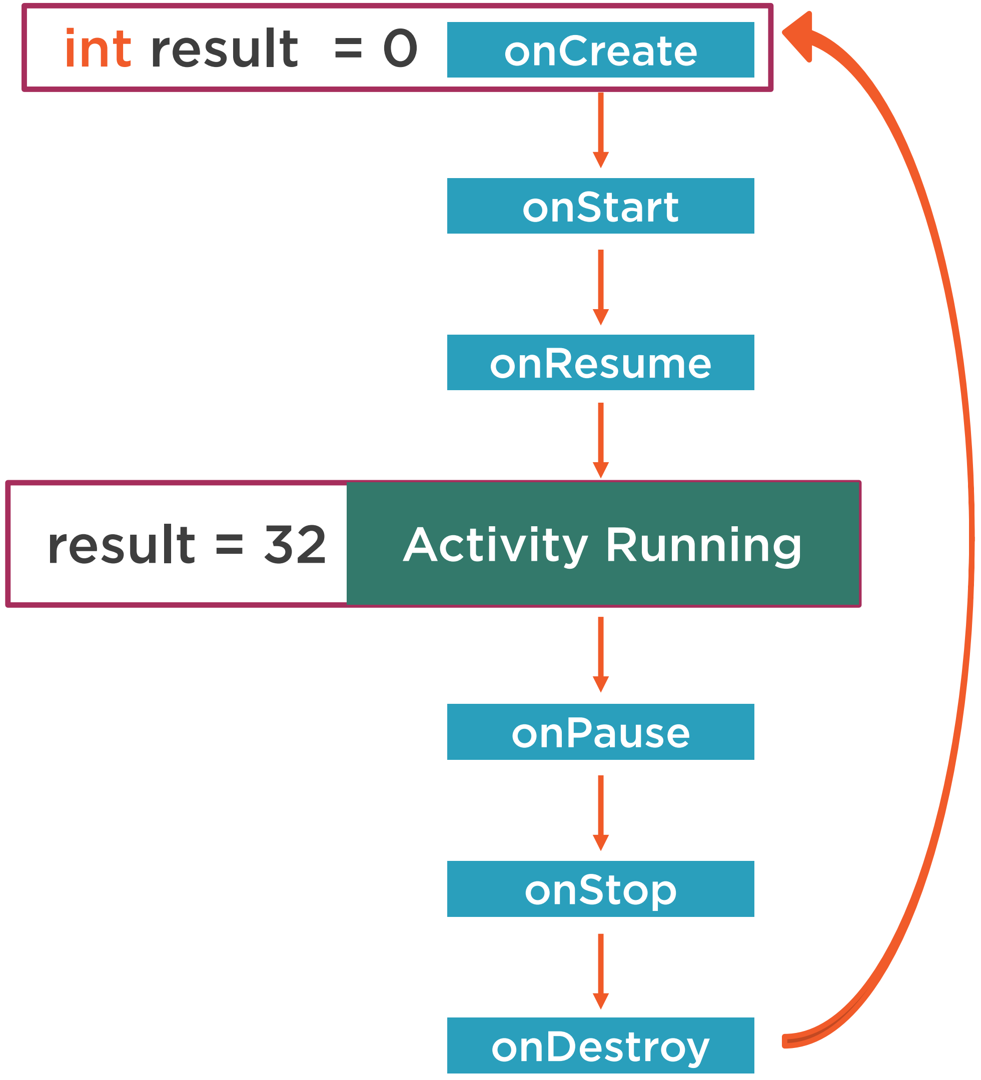
On rotation of screen

- The Activity is Destroyed
- The Activity is Recreated fresh in requested orientation

Impact on Views and Variables?

Consequences of Screen Orientation

Activity and Screen Rotation



int result = 0

onCreate

onStart

onResume

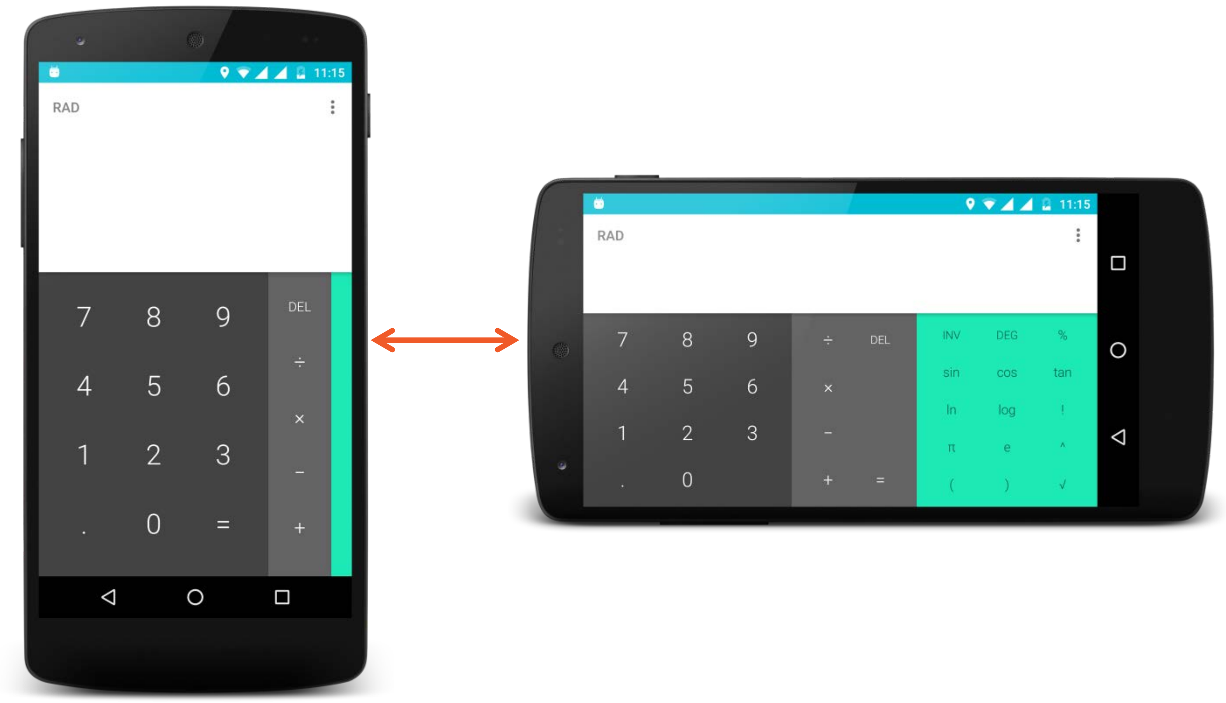
result = 32

Activity Running

onPause

onStop

onDestroy



Impact of Screen Orientation on Views

Summary

[Provided Views have unique ID]

Views that DO NOT RETAIN their state on screen rotation

- TextView
- Button

Views that by default RETAIN their state on screen rotation

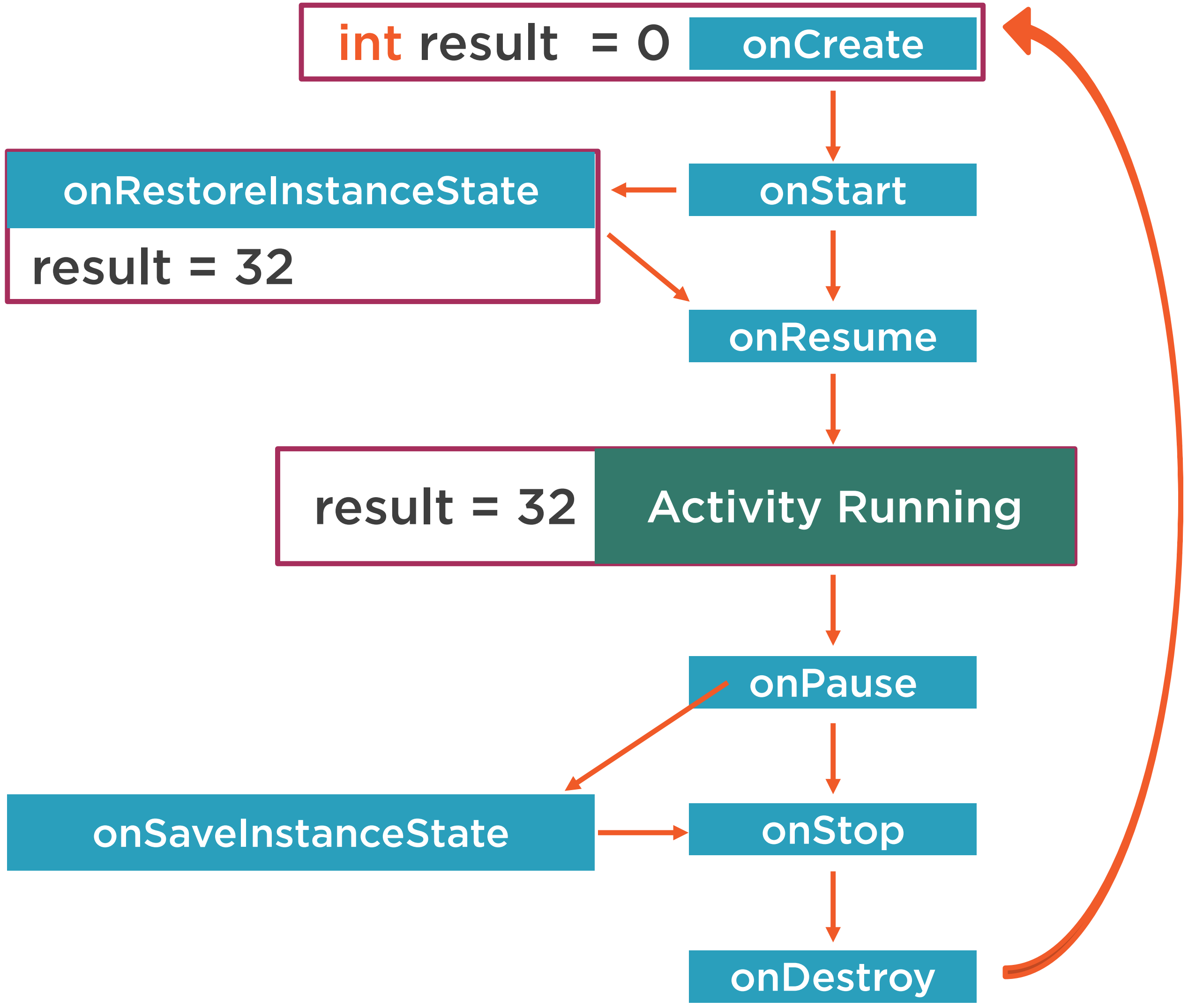
- EditText
- Checkbox
- Switch
- RadioButton

Solutions

Possible Solutions to handle Screen Orientation

1. **Restoring the Activity State**
   - Activity is destroyed and recreated
   - Use onRestoreInstanceState and onSaveInstanceState
2. **Handling configuration change yourself**
   - Activity is not destroyed
   - Use onConfigurationChanged

1. Restoring the Activity State



int result = 0

onCreate

onStart

onRestoreInstanceState

result = 32

onResume

result = 32

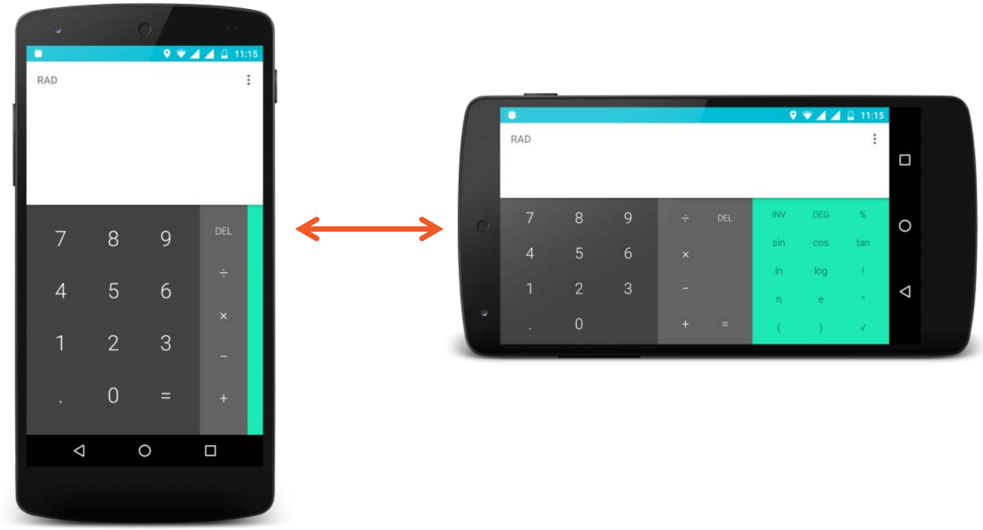
Activity Running

onPause

onSaveInstanceState

onStop

onDestroy

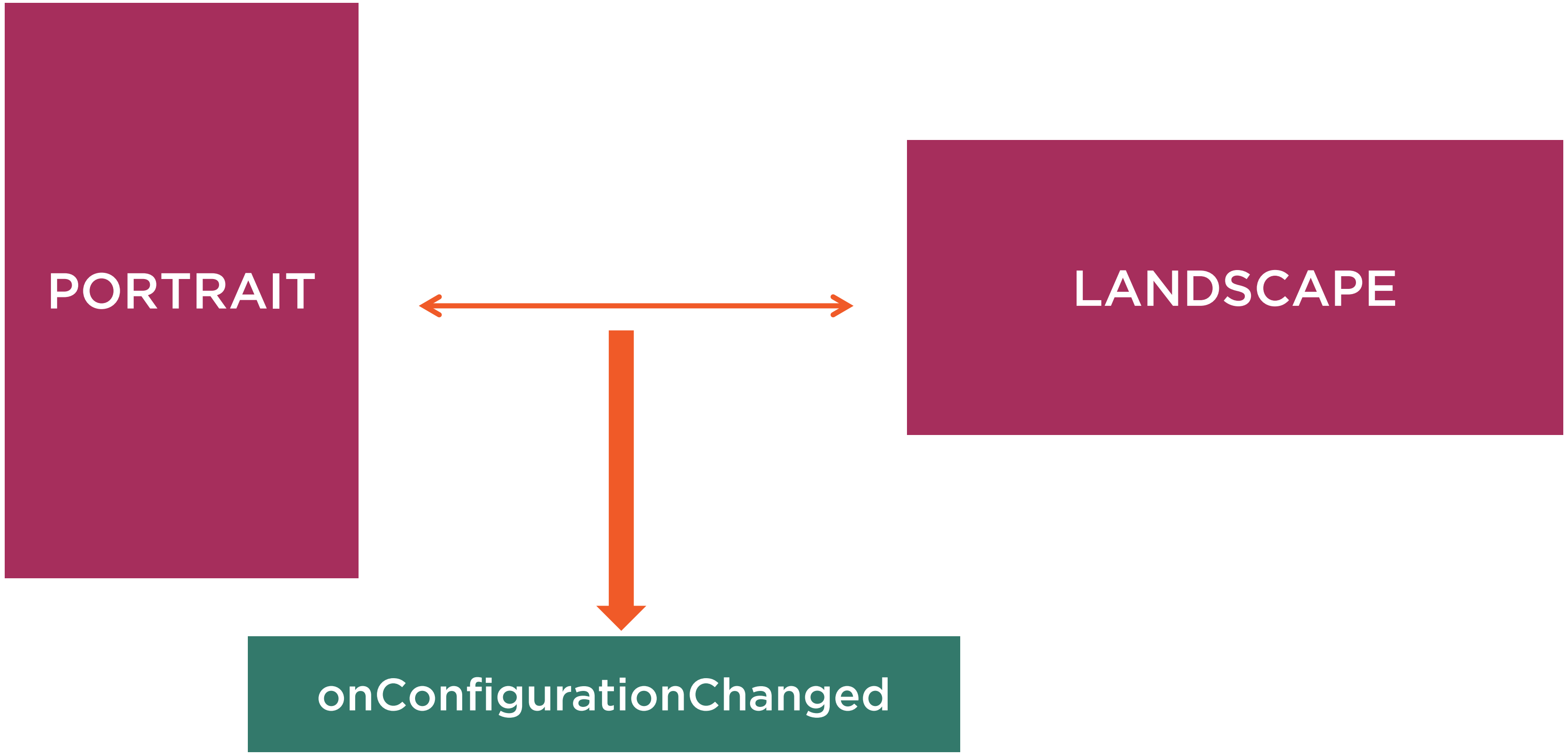


Demo

Handling Screen Orientation

1. Restoring the Activity State
2. Handling Configuration Change Yourself

2. Handling Configuration Change Yourself



1. Declare android:configChanges attribute in Manifest file under <activity>
2. Override onConfigurationChanged() in Activity class

Summary

When Screen is rotated

- Activity is destroyed
- Activity is recreated in requested orientation
- Views and Variables needs to be handled

Summary

Possible Solutions to handle Screen Orientation

1. **Restoring the Activity State using onRestoreInstanceState and onSaveInstanceState**
   - Is generally preferred and recommended
2. **Handling configuration change yourself using onConfigurationChanged**
   - Against the guidelines
   - Not recommended
   - Should be used ONLY when restoring Activity back is expensive and slow

What Next?

Android Fragments

Other Android Components

- BroadcastReceiver
- Service
- ContentProvider