Create a new Button Programmatically in Kotlin Android

Create EditText Programmatically

We know that we can specify Button widget using layout file. But, we can also create a Button programmatically and then add this to a specific View as child, in layout file.

In this tutorial, we will learn how to create a Button widget programmatically in Android, and add this Button to a LinearLayout in layout file.

Code – Create EditText in Kotlin File

A quick snippet of code to create a new Button programmatically in Kotlin Android

```
val button_dynamic = Button(this)
button_dynamic.layoutParams = LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT
button_dynamic.text = "Dynamic Button"
```

Creating a new Button programmatically at a point in the program requires it to be in UI thread. And also we need the application context to create any new View. Android prevents any View to be created outside the UI thread by throwing a build error.

In this Android Tutorial, we shall learn how to create a Button programmatically and add the Button to a LinearLayout using Kotlin.

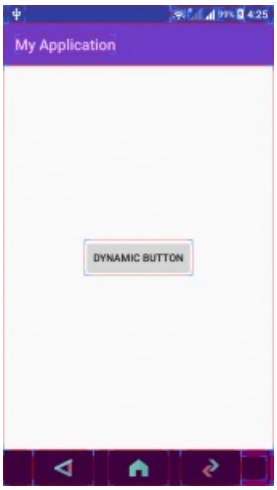
activity_main.xml: Following is the activity_main.xml containing an empty LinearLayout to which we shall add the dynamic Button.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res
    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"
    tools:context="com.tutorialkart.myapplication.MainActivity">
    <LinearLayout
        android:id="@+id/ll_main_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical">
    </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

MainActivity.kt: We shall create a new Button with text “Dynamic Button” and add it to the LinearLayout. This addition makes the dynamically created Button to be appended at the end of all child views present in the LinearLayout.

```
package com.tutorialkart.myapplication

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.view.ViewGroup
import android.widget.Button
import android.widget.LinearLayout

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val ll_main = findViewById(R.id.ll_main_layout) as LinearLayout

        // creating the button
        val button_dynamic = Button(this)
        // setting layout_width and layout_height using layout parameters
        button_dynamic.layoutParams = LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP
        button_dynamic.text = "Dynamic Button"
        // add Button to LinearLayout
        ll_main.addView(button_dynamic)
    }
}
```

Following is the Output with layout bounds :



Create a new Button programmatically in Kotlin Android

Conclusion

In this Kotlin Android Tutorial, we have learned to create Button programmatically and add it to layout.

- Kotlin vs Java
- Use Java 8 in Android
- Add External Jar to Android Dependencies

Android TextView

- Android TextView
- Android TextView - Basic Example
- Android TextView - Create programmatically
- Android TextView - OnClickListener
- Android TextView - Justify Text
- Android TextView - Italic
- Android TextView - Bold

Android Button

- Android - New Button programmatically
- Android Button - OnClickListener
- Android Button - Disable All Caps
- Android Button - Custom Background
- Android Button - Change background programatically

Android Toast

- Android Toast - Example

Android EditText

- Android EditText - Create programmatically
- Android EditText - On Text Change - Listener
- Android TextInputLayout - Floating Label in EditText
- Android EditText - Keyboard with only Numbers
- Android EditText - Show/Hide Password

Android ImageView

- Android ImageView - OnClickListener

Android Radio Buttons

- Android RadioGroup - RadioButtons Create programmatically

Android SeekBar

- Android SeekBar - Example
- Android SeekBar Set Custom Range

Android Intent

- Android - Start Another Activity
- Android - Open URL in Browser Activity

Android AlertDialog

- Android AlertDialog - Example

Android WebView

- Android WebView - Example

Android ProgressBar

- Kotlin Android - Inderterminate ProgressBar

Android Snackbar

- Android Snackbar - Example
- Android Snackbar - Set Action
- Android Snackbar - Change Text Color, Background Color

Android ListView

- Android ListView Example
- Android Refresh ListView

Android Device Parameters

- Android Get Screen Width and Height Programatically

Android Canvas

- Draw Rect / Oval to Canvas
- Android Draw Circle Border
- Android Draw SVG to Canvas

Android Programming - Other

- Android - Access View Programmatically using findViewById
- Android runOnUiThread

Android Game Development

- Android Game Development
- Detect Collisions between two Sprites (Bitmaps)

Android Text To Speech

- Android Text To Speech - Kotlin Example

Fix Errors

- Android - Minimum supported Gradle version

- Android - All support libraries must use the exact same version specification

Example Applications

- Android - Login Form
- Android - Color Picker
- Kotlin Android Game Development

Kotlin - Java

- Kotlin Tutorial

Useful Resources

- How to Learn Programming