

Android changing Floating Action Button color

Asked 6 years, 2 months ago Active 1 year ago Viewed 412k times



I have been trying to change Material's Floating Action Button color, but without success.

604



96



```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/profile_edit_fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end|bottom"
    android:layout_margin="16dp"
    android:clickable="true"
    android:src="@drawable/ic_mode_edit_white_24dp" />
```

I have tried to add:

```
android:background="@color/mycolor"
```

or via code:

```
FloatingActionButton fab = (FloatingActionButton)
rootView.findViewById(R.id.profile_edit_fab);
fab.setBackgroundColor(Color.parseColor("#mycolor"));
```

or

```
fab.setBackgroundDrawable(new ColorDrawable(Color.parseColor("#mycolor")));
```

But none of the above worked. I have also tried the solutions in the proposed duplicate question, but none of them works; the button remained green and also became a square.

P.S. It would be also nice to know how to add ripple effect, couldn't understand that either.

[java](#) [android](#) [android-5.0-lollipop](#) [floating-action-button](#)

Share Improve this question

Follow

edited Feb 15 '20 at 15:24



halfer

18.7k

15

80

163

asked Jun 21 '15 at 21:20



Jjang

10.4k

10

44

76

- 2 possible duplicate of [Change color of Floating Action Button from Appcompat 22.2.0 programmatically.](#) – [Jared Burrows](#) Jun 21 '15 at 21:57

Ripple effect is not available on pre-lollipop devices because it utilizes a new [RenderThread](#). – [tachyonflux](#) Jun 21 '15 at 22:00

@karaokyo ok but how am I doing it? – [Jjang](#) Jun 22 '15 at 17:20

- 41 Google does a real bad job in my opinion to make these things accesible – [Joop](#) Jul 17 '15 at 13:51

To do this programmatically and backwards compatible, see stackoverflow.com/questions/30966222/...
 – Ralph Pina Jul 27 '16 at 16:01

25 Answers

Active	Oldest	Votes
--------	--------	-------

▲
1181

As described in the [documentation](#), by default it takes the color set in **styles.xml** attribute **colorAccent**.

▼
✓

The background color of this view defaults to the your theme's colorAccent. If you wish to change this at runtime then you can do so via `setBackgroundTintList(ColorStateList)`.



If you wish to change the color

- in XML with attribute **app:backgroundTint**

```
<android.support.design.widget.FloatingActionButton
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_add"
    app:backgroundTint="@color/orange"
    app:borderWidth="0dp"
    app:elevation="6dp"
    app:fabSize="normal" >
```

- in code with **.setBackgroundTintList** (answer below by [ywwynm](#))

As @Dantalian mentioned in the comments, if you wish to change the icon color for Design Support Library **up to v22 (inclusive)**, you can use

```
android:tint="@color/white"
```

For Design Support Library **since v23** for you can use:

```
app:tint="@color/white"
```

Also with `androidx` libraries you need to set a 0dp border in your xml layout:

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_add"
    app:backgroundTint="@color/orange"
    app:borderWidth="0dp"
    app:elevation="6dp"
    app:fabSize="normal" />
```

Share Improve this answer

edited Jan 15 '20 at 8:52

answered Jul 2 '15 at 8:24

Follow



Marko

19.6k

13

45

61

- 24 This doesn't work in Design Support Library v23 for < API 22 devices. It was working in v22, so they broke something, apparently. – [hungryghost](#) Oct 6 '15 at 5:39
- 5 Note that if you want to also change the image color to white for example you can use:
`android:tint="@android:color/white"` – [Dantalian](#) Nov 4 '15 at 11:11
- 1 `android:tint="@color/white"` didnt work for me because it couldn't resolve it for some reason.
`android:tint="#ffffff"` Works for me though – [realappie](#) Nov 6 '15 at 2:02
- 2 That is because `@color/white` does not exists in your colors.xml. use `@android:color/white` or create a color named white or whatever you need – [Dantalian](#) Nov 12 '15 at 8:40
- 244 use `"app:backgroundTint"` not `"android:backgroundTint"` – [Vasil Valchev](#) Feb 4 '16 at 8:47



256



Vijet Badigannavar's answer is correct but using `colorStateList` is usually complicated and he didn't tell us how to do it. Since we often focus on changing `view`'s color in normal and pressed state, I'm going to add more details:

1. If you want to change `FAB`'s color in normal state, you can just write

```
mFab.setBackgroundTintList(ColorStateList.valueOf(your_color_in_int));
```

2. If you want to change `FAB`'s color in pressed state, thanks for Design Support Library **22.2.1**, you can just write

```
mFab.setRippleColor(your_color_in_int);
```

By setting this attribute, when you long-pressed the `FAB`, a ripple with your color will appear at your touch point and reveal into whole surface of `FAB`. Please notice that it won't change `FAB`'s color in normal state. Below API 21(Lollipop), there is no ripple effect but `FAB`'s color will still change when you're pressing it.

Finally, if you want to implement more complex effect for states, then you should dig deeply into `ColorStateList`, here is a SO question discussing it: [How do I create ColorStateList programmatically?](#)

UPDATE: Thanks for @Kaitlyn's comment. To remove stroke of FAB using `backgroundTint` as its color, you can set `app:borderWidth="0dp"` in your xml.

Share Improve this answer Follow

edited May 23 '17 at 10:31



Community ♦

1 1

answered Aug 16 '15 at 1:27



ywynm

10.9k

7

33

52

- 3 Thanks for this, I didn't know how to use `ColorStateList`. However, your code leaves my FAB with a stroke (outline) of the theme accent color. Do you have any idea how I can remove that stroke (or change it's color, rather) and have a FAB in a special color that looks like that special color is the accent color? – [Kaitlyn Hanrahan](#) Aug 20 '15 at 15:08

@KaitlynHanrahan Please see my update. Thanks for your comment. – [ywynm](#) Aug 21 '15 at 9:12

That works perfect -- Lollipop and Kitkat look the same, even both have that little shading so the FAB pops against the same color in the background (I have it partially over the Toolbar). Thanks so much! It's simple but I don't know if I ever would have found that on my own. – [Kaitlyn Hanrahan](#) Aug 22 '15 at 17:21

Is there a way how to set `app:borderWidth` programatically? Thanks – [Štarke](#) Oct 22 '15 at 6:50

@Štarke Well, after looking at the source code of `FloatingActionButton`, I'm afraid that you cannot set it programatically right now since there is no setter for `mBorderWidth`. – [ywwynm](#) Oct 22 '15 at 13:07



138



As [Vasil Valchev](#) noted in a comment it is simpler than it looks, but there is a subtle difference that I wasn't noticing in my XML.

```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/profile_edit_fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end|bottom"
    android:layout_margin="16dp"
    android:clickable="true"
    android:src="@drawable/ic_mode_edit_white_24dp"
    app:backgroundTint="@android:color/white"/>
```

Notice it is:

```
app:backgroundTint="@android:color/white"
```

and **not**

```
android:backgroundTint="@android:color/white"
```

Share Improve this answer Follow

edited Apr 18 '18 at 13:51



[Taslim Oseni](#)

4,743

10

35

50

answered Feb 12 '16 at 11:09



[HenriqueMS](#)

3,381

2

26

36

8 This is important because these are two completely different things: `android:backgroundTint=""` is from *Api level 21* and part of the Lollipop Material Design and will be ignored below Lollipop. Also it doesn't change the FAB colour completely because it is not a solid color. You have to use the `app:` one to make it work. – [creativecreatorormaybenot](#) Feb 5 '17 at 15:33

1 Thanks, using `android:color` caused my app to crash, and it was driving me nuts! Thanks for saving my app and what's left of my sanity ;-)) – [Kapenaar](#) Mar 30 '17 at 17:47

@Kapenaar no problem, was scratching my head for a while as well before noticing ;) – [HenriqueMS](#) Mar 31 '17 at 12:56

How would this work programatically since on the java side there is no difference like `app:` and `android:` in the XML. – [marienke](#) May 22 '17 at 7:58

@marienke you can set it with a `ColorStatesList`, post a new question and post the link here so I can answer it ;) – [HenriqueMS](#) May 22 '17 at 8:31



if you try to change color of FAB by using app, there some problem. frame of button have different color, so what you must to do:

62



```
app:backgroundTint="@android:color/transparent"
```



and in code set the color:

```
actionButton.setBackgroundTintList(ColorStateList.valueOf(getResources().getColor(R.color
```

Share Improve this answer Follow

answered Sep 16 '15 at 14:07



[ergo](#)

1,068

8

6

2 This work for normal colors. If I have given alpha color like #44000000 the frame color issue is reproducible. – [Dinesh](#) Feb 12 '16 at 9:38

5 getResources.getColor() is deprecated, consider using ContextCompat.getColor(context, your colour); instead. Hope it helps – [Tinashe Chinyanga](#) May 6 '16 at 9:24



just use,

54



```
app:backgroundTint="@color/colorPrimary"
```



dont use,

```
android:backgroundTint="@color/colorPrimary"
```

Share Improve this answer Follow

answered Mar 8 '16 at 9:00



[Ashana.Jackol](#)

2,706

25

20

hmmmm this cause android studio autocomplete give result use android:backgroundTint , this answer very good – [Yudi karma](#) Apr 25 '19 at 7:31



The FAB is colored based on your `colorAccent` .

33



```
<style name="AppTheme" parent="Base.Theme.AppCompat.Light">
  <item name="colorAccent">@color/accent</item>
</style>
```



Share Improve this answer Follow

answered Jun 21 '15 at 21:55



[tachyonflux](#)

19.4k

7

45

66


Well, the default color is not green. Where did you set a green color? – [tachyonflux](#) Jun 22 '15 at 17:44

1 I did not. As you see, this xml is the only place I set up the layout of the button... – [Jjang](#) Jun 22 '15 at 17:50


Maybe you should create a new project with just the FAB and post the code for the entire thing. – [tachyonflux](#) Jun 22 '15 at 17:59

3 Wow... I basically just asked you to do the same thing and you weren't willing to do it. It is quite hypocritical of you to believe that your code was adequate, but that my addition to your code is not adequate. It falls to the asker to create an [MCVE](#). Thanks. – [tachyonflux](#) Jun 22 '15 at 18:27

3 Not anymore, based on [Theme Attribute Mapping](#) the Floating Action Button now (I verified myself) is colored to **colorSecondary** (*com.google.android.material:material:1.1.0-alpha02*) – [ievgen](#) Feb 4 '19 at 7:26

 `mFab.setBackgroundTintList(ColorStateList.valueOf(ContextCompat.getColor(mContext,R.color`

27

 Share Improve this answer Follow

edited Jun 16 '17 at 12:50

answered Jun 16 '17 at 11:22



[atoMerz](#)

7,018 15 57 99



[Reza Isfandyari](#)

271 3 2

This is the easiest solution, way easier than using a custom drawable. – [IgorGanapolsky](#) Nov 14 '17 at 20:55

 New theme attribute mapping for [Floating Action Button](#) in material 1.1.0

18

 In your app theme:

- Set `colorSecondary` to set a color for background of FAB (maps to `backgroundTint`)
- Set `colorOnSecondary` to set a color for icon/text and ripple color of FAB (maps to `tint` and `rippleColor`)

```
<style name="AppTheme" parent="Theme.MaterialComponents.Light.NoActionBar">
    <!-- ...whatever else you declare in your app theme.. -->
    <!-- Set colorSecondary to change background of FAB (backgroundTint) -->
    <item name="colorSecondary">@color/colorSecondary</item>
    <!-- Customize colorSecondary to change icon/text of FAB (maps to tint and
    rippleColor) -->
    <item name="colorOnSecondary">@android:color/white</item>
</style>
```

Share Improve this answer Follow

answered May 15 '19 at 23:33




[luca992](#)

1,396 19 25

Finally someone who provides simple solutions for FAB background and foreground (icon) colors. – [ManuelITS](#) Jan 5 '20 at 15:16

Thanks. That's what I was looking for – [Mateus Melo](#) Jul 19 '20 at 14:41

for more info: material.io/develop/android/theming/color + [material.io/design/color/...](https://material.io/design/color/) – [jacoballenwood](#) Jun 21 at 21:50 



17

Other solutions may work. This is the 10 pound gorilla approach that has the advantage of being broadly applicable in this and similar cases:

Styles.xml:



```
<style name="AppTheme.FloatingActionButtonOverlay" >
    <item name="colorAccent">@color/colorFloatingActionBarAccent</item>
</style>
```

your layout xml:

```
<android.support.design.widget.FloatingActionButton
    android:theme="AppTheme.FloatingActionButtonOverlay"
    ...
</android.support.design.widget.FloatingActionButton>
```


Share Improve this answer Follow

answered Nov 21 '15 at 15:10




[Robin Davies](#)

6,976 1 29 41

Hmm, seems like this should work, but I'm getting compiler errors in the layout file String types not allowed (at 'theme' with value 'AppTheme.FloatingActionButtonOverlay') . Maybe I'm missing a point with the styles.xml.... – [SMBiggs](#) Aug 25 '16 at 6:30 

@ScottBiggs use android:theme="@style/AppTheme.FloatingActionButtonOverlay" but this solution does not work for me either way... – [jpact](#) Jan 14 '17 at 21:20

What about API 16 which does not have colorAccent attribute? – [Dushyant Suthar](#) Apr 23 '17 at 9:51 



14

The document suggests that it takes the @color/accent by default. But we can override it on code by using

```
fab.setBackgroundTintList(ColorStateList)
```



Also remember,

The minimum API version to use this library is 15 so you need to update it! if you don't want to do it then you need to define a custom drawable and decorate it!

Share Improve this answer Follow

edited Jun 29 '15 at 18:32

answered Jun 24 '15 at 18:05



[Vijet Badigannavar](#)

522 3 15

works, but then I have to target api 21+. if I target api 8 can't call this method. – [Jjang](#) Jun 26 '15 at 15:12

Just include this design library that has been released by google "com.android.support.design:22.2.0". This can be used on devices that are non-lollipop versions! cheers!! – [Vijet Badigannavar](#) Jun 27 '15 at 18:10 

of course I do, but this is the error on this code: Call requires API level 21 (current min is 8):
android.widget.ImageView#setBackgroundTintList – [Jjang](#) Jun 27 '15 at 18:23

The minimum API version to use this library is 15 so you need to update it! if you dont want to do it then you need to define a custom drawable and decorate it! – [Vijet Badigannavar](#) Jun 27 '15 at 18:53

Oh thanks! I didn't know, thought it was 21 since this is what the error said. Please post this as an answer and I'll accept it! – [Jjang](#) Jun 28 '15 at 17:29



14



With the [Material Theme](#) and the **material components** [FloatingActionButton](#) by default it takes the color set in `styles.xml` attribute `colorSecondary`.

- You can use the `app:backgroundTint` attribute in xml:

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    ...
    app:backgroundTint=".."
    app:srcCompat="@drawable/ic_plus_24"/>
```

- You can use `fab.setBackgroundTintList();`
- You can customize your style using the `<item name="backgroundTint">` attribute

```
<!--<item
name="floatingActionButtonStyle">@style/Widget.MaterialComponents.FloatingActionButton</i
-->
<style name="MyFloatingActionButton"
parent="@style/Widget.MaterialComponents.FloatingActionButton">
    <item name="backgroundTint">#00f</item>
    <!-- color used by the icon -->
    <item name="tint">@color/...</item>
</style>
```

- starting from version **1.1.0** of material components you can use the new `materialThemeOverlay` attribute to override the default colors only for some components:

```
<style name="MyFloatingActionButton"
parent="@style/Widget.MaterialComponents.FloatingActionButton">
    <item name="materialThemeOverlay">@style/MyFabOverlay</item>
</style>

<style name="MyFabOverlay">
    <item name="colorSecondary">@color/custom2</item>
    <!-- color used by the icon -->
    <item name="colorOnSecondary">@color/...</item>
</style>
```


[Share](#) [Improve this answer](#) [Follow](#)

edited Oct 13 '19 at 22:00

answered Aug 28 '19 at 9:26

**Gabriele Mariotti****208k**

63

519

540



Changing Floating action button background color by using below line

12

```
app:backgroundTint="@color/blue"
```



Changing Floating action button icon color

```
android:tint="@color/white"
```

[Share](#) [Improve this answer](#) [Follow](#)

edited Nov 15 '18 at 13:27

answered Oct 12 '17 at 7:23

**Agilanbu****2,398**

1

26

29

**Kishore Reddy****2,120**

17

15



Thanks to autocomplete. I got lucky after a few hit and trials:

9

```
xmlns:card_view="http://schemas.android.com/apk/res-auto"
card_view:backgroundTint="@color/whicheverColorYouLike"
```



-- or -- (both are basically the same thing)

```
xmlns:app="http://schemas.android.com/apk/res-auto"
app:backgroundTint="@color/whicheverColorYouLike"
```

This worked for me on API Version 17 with design library 23.1.0.

[Share](#) [Improve this answer](#) [Follow](#)

answered Nov 16 '15 at 9:30

**Gaurav Kumar Gupta****671**

7

7



I got the same problem and its all snatching my hair. Thanks for this

<https://stackoverflow.com/a/35697105/5228412>

9



What we can do..



```
favourite_fab.setImageDrawable(ContextCompat.getDrawable(getBaseContext(),
R.drawable.favourite_selected));
```

it works fine for me and wish for others who'll reach here.

Share Improve this answer Follow

edited May 23 '17 at 11:47



Community ♦

1 1

answered Mar 22 '16 at 17:44



Purvik Rana

193 2 11

Whilst this may theoretically answer the question, [it would be preferable](#) to include the essential parts of the answer here, and provide the link for reference. – Bhargav Rao ♦ Mar 22 '16 at 17:51



9



```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    app:elevation="6dp"
    app:backgroundTint="@color/colorAccent"
    app:pressedTranslationZ="12dp"
    android:layout_margin="@dimen/fab_margin"
    android:src="@drawable/add"/>
```

Note that you add colors in res/values/color.xml and include the attribute in your fab

```
app:backgroundTint="@color/addedColor"
```

Share Improve this answer Follow

answered Jun 21 '16 at 7:07



Gavine Joyce

351 3 8



6



When using [Data Binding](#) you can do something like this:

```
android:backgroundTint="@{item.selected ? @color/selected : @color/unselected}"
```

I have made a very simple [example](#)

Share Improve this answer Follow

edited Oct 12 '18 at 12:42

answered Sep 13 '18 at 8:50



Jan Málek

454 6 16



4



i did it like this `android:background="@color/colorAccent"` i just go to folder res then click on folder values and then on colors.xml in colors.xml I just change the color of colorAccent and call it in `android:background` and its done

Share Improve this answer Follow

answered Apr 29 '18 at 16:46



sana sajjad

41 1



If you have a Floating Action Button with no drawable you can change the tint programmatically using:

4



```
fab.setBackground().mutate().setTint(ContextCompat.getColor(yourContext,
R.color.anyColor));
```



Share Improve this answer Follow

answered Mar 6 '20 at 12:30



Tom

41 1

Thats the solution I was searchin for. I had a simple fab with backgroundTint and couldn't find a way to change it programmatically until I found this. – Redar Apr 3 '20 at 16:42

I use Data Binding, and I got a NullPointerException... – inspire_coding Aug 26 '20 at 15:19



3



The point we are missing is that before you set the color on the button, it's important to work on the value you want for this color. So you can go to values > color. You will find the default ones, but you can also create colors by copying and pasting them, changing the colors and names. Then... when you go to change the color of the floating button (in activity_main), you can choose the one you have created



Exemple - code on values > colors with default colors + 3 more colors I've created:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>

    <color name="corBotaoFoto">#f52411</color>
    <color name="corPar">#8e8f93</color>
    <color name="corImpar">#494848</color>

</resources>
```

Now my Floating Action Button with the color I've created and named "corPar":

```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="@dimen/fab_margin"
    android:src="@android:drawable/ic_input_add"
    android:tint="#ffffff"
    app:backgroundTint="@color/corPar"/>
```

It worked for me. Good Luck!

Share Improve this answer Follow

answered Apr 15 '16 at 17:16

Nidia F.



31 1



3



for Material design, I just changed the floating action button color like this, Add the below two lines in your Floating action button xml. And done,

```
android:backgroundTint="@color/colorPrimaryDark"
app:borderWidth="0dp"
```



Share Improve this answer Follow

answered Apr 25 '20 at 7:18



Jwala Kumar

384 3 5



2



My solution, which worked for me with Data Binding

```
val color = ContextCompat.getColor(context, R.color.colorPrimary)
binding.fab.backgroundTintList = ColorStateList.valueOf(getColor)
```



Share Improve this answer Follow

answered Aug 26 '20 at 15:25



inspire_coding

251 3 10



1



in Kotlin:

```
val gray = getColor(requireContext(), R.color.green)
binding.fabSubmit.backgroundTintList = ColorStateList.valueOf(gray)
```



Share Improve this answer Follow

answered Jun 8 '20 at 19:44



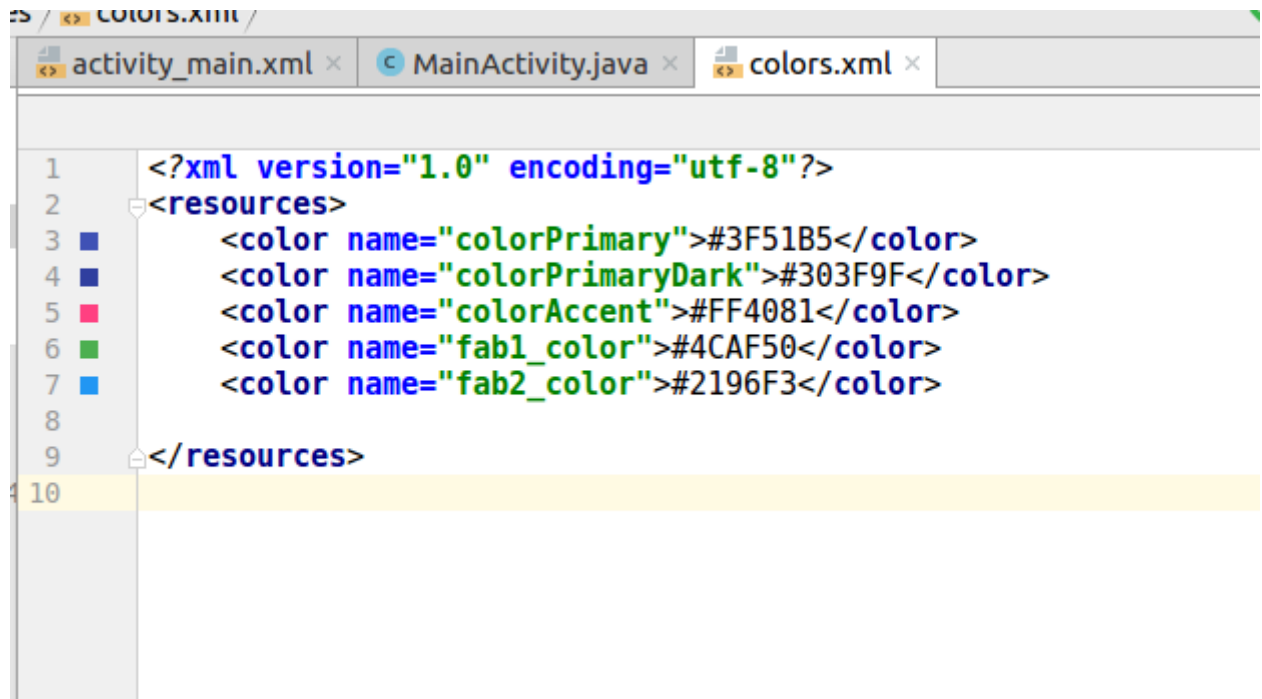
fullmoon

6,765 3 37 53



0





add colors in color.xml file and then add this line of code...

```
floatingActionButton.setBackgroundTintList(ColorStateList.valueOf(getResources().getColor(R.color.fab2_color)));
```

Share Improve this answer Follow

answered Aug 31 '18 at 16:32



[Progga Ilma](#)

410 3 6

You can use this code in case you want to change the color programmatically

0

```
floating.setBackgroundTintList(getResources().getColorStateList(R.color.vermelho));
```

Share Improve this answer Follow

edited Aug 12 '19 at 10:48

answered Jun 7 '18 at 19:11



[OhhhThatVarun](#)

2,625 1 15 36



[Diego Souza](#)

9 1

use

0

app:backgroundTint="@color/orange" in

```

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/id_share_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/share"
    app:backgroundTint="@color/orange"
    app:fabSize="mini"
    app:layout_anchorGravity="end|bottom|center" />

```

```
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

[Share](#) [Improve this answer](#) [Follow](#)

answered Sep 1 '19 at 12:00

[passionatedevops](#)

389 2 5



Highly active question. Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.