



Профиль

Публикации (1)

Комментарии (6)

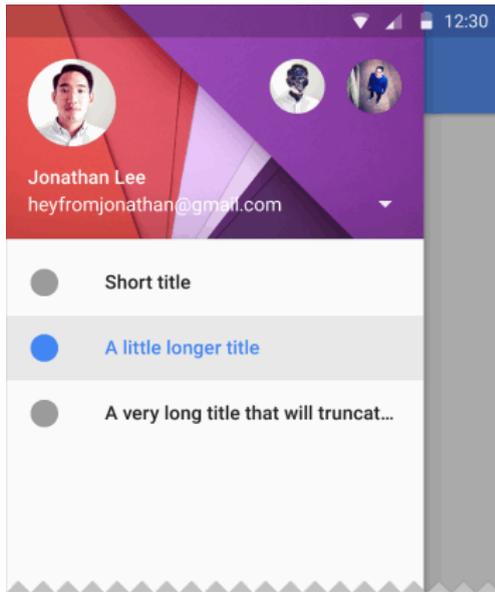
Избранное (24)

16 февраля в 22:57

## Navigation Drawer в стиле Material Design за 5 минут

из песочницы

Разработка под Android\*, Mobile Development\*



В данной статье я расскажу, как быстро добавить в ваше приложение для Android боковое меню (aka Navigation Drawer) в стиле Material Design. Для этого мы воспользуемся [библиотекой](#), любезно предоставленной [Mike Penz](#).

У вас получится Navigation Drawer, который:

- Соответствует последним рекомендациям по дизайну (Google Material Design Guidelines);
- Поддерживает использование нескольких Drawer (можно выдвигать второй справа);
- Поддерживает использование бейджей;
- Имеет простой и понятный интерфейс (API);
- Может вылезать как под, так и поверх Status Bar;
- Позволяет менять иконки, цвета, бейджи во время выполнения;
- Использует AppCompat support library;
- Работает, начиная с API 14.

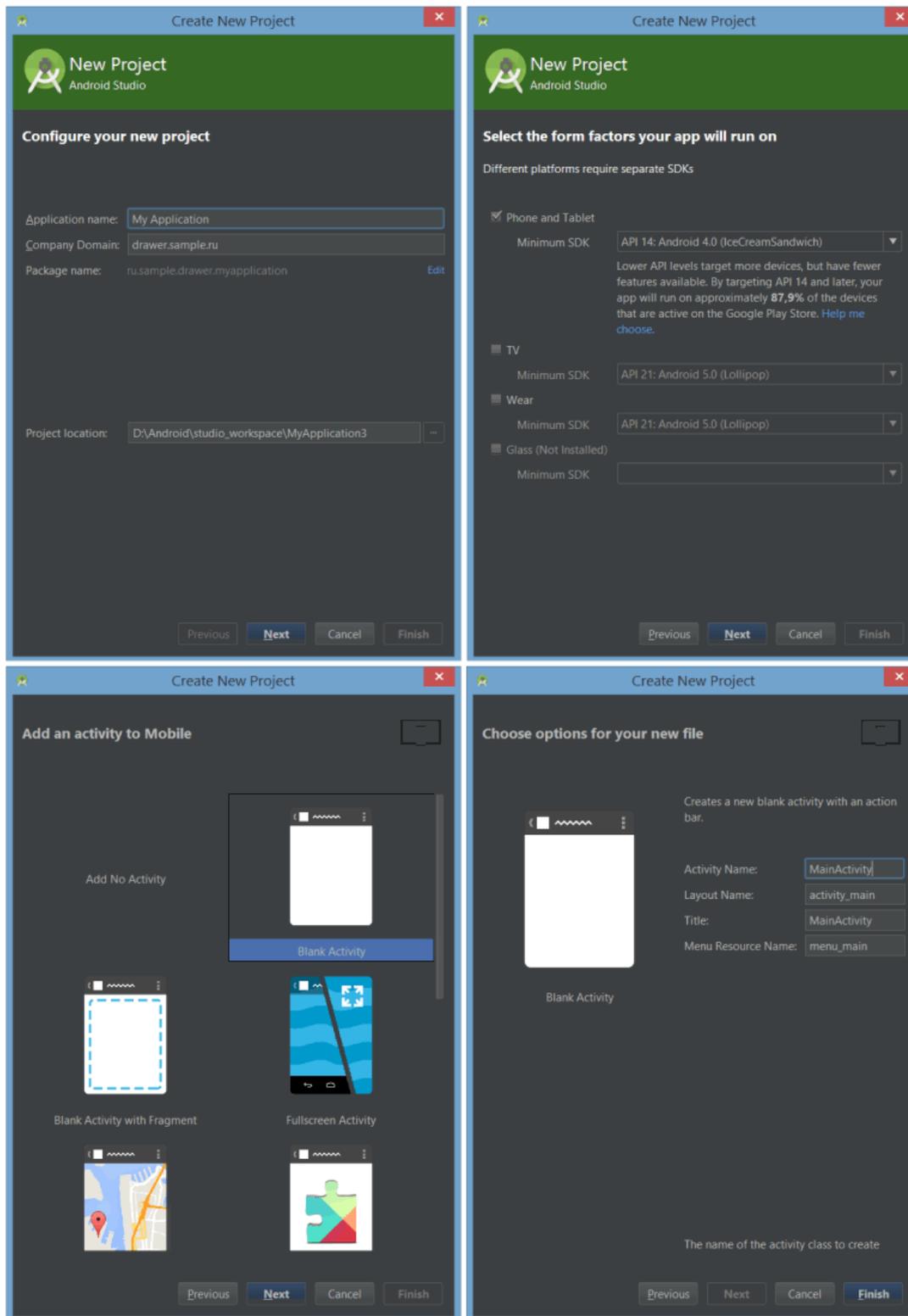
Помимо этого, новички обучатся интеграции сторонних библиотек в свой проект, что крайне полезно, учитывая их

грандиозное разнообразие на Github.

### Создание проекта

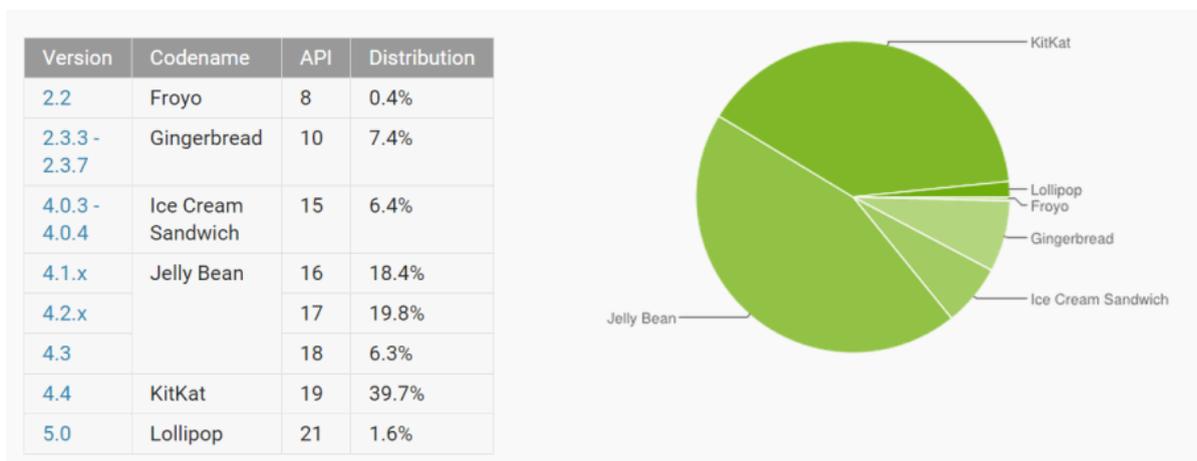
В примере будет использоваться интегрированная среда разработки Android Studio от компании Google, основанная на IntelliJ IDEA, которую сама корпорация активно продвигает. Все действия можно воспроизвести используя и другие среды, например, Eclipse. Однако статья ориентирована на новичков, а они будут в большинстве своем использовать именно Android Studio, так как именно его Google теперь и предлагает при скачивании Android SDK с [developer.android.com](http://developer.android.com) (ранее можно было скачать Eclipse).

Итак, выбираем в меню «File» -> «New Project...»:



Заполняем имя приложения, пакета, выбираем SDK.

Создавать проект мы будем с поддержкой минимального API Level равного 14, что соответствует Android 4.0 Ice Cream Sandwich, поскольку всё, что ниже, составляет менее 8% аудитории и приносит несоизмеримо большее количество головной боли:



В последних двух окнах оставляем все по умолчанию, жмем «Finish».

## Android Support Library

Для того, чтобы красивый Navigation Drawer работал на версиях Android ниже 5.0 и выглядел в стиле Material Design, необходимо включить в проект библиотеку поддержки от Google, которая носит название [v7 appcompat library](#). В текущей версии Android Studio (1.0.2) библиотека подключается по умолчанию при создании проекта. Проверьте это в файле проекта `\app\build.gradle`, в разделе `dependencies` должна быть строка (цифры могут быть не обязательно «21.0.3»):

```
compile 'com.android.support:appcompat-v7:21.0.3'
```

а класс `MainActivity` должен наследоваться от `ActionBarActivity`

```
public class MainActivity extends ActionBarActivity {
```

Также проверьте в `\res\values\styles.xml`, чтобы тема приложения наследовалась от `Theme.AppCompat` или ее вариаций без `ActionBar` (мы заменим `ActionBar` на `ToolBar`), например:

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
```

## Подключение библиотеки MaterialDrawer

Добавьте в раздел `dependencies` файла `\app\build.gradle` строки

```
compile('com.mikepenz.materialdrawer:library:0.9.5@aar') {
    transitive = true
}
```

и нажмите появившуюся в верхней части окна кнопку «Sync Now» для синхронизации вашего проекта.

## Подготовка разметки для Navigation Drawer

В главный `layout` приложения нужно добавить `ToolBar`. Приведите `activity_main.xml` к такому виду:

```
<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">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:elevation="4dp"
        android:minHeight="?attr/actionBarSize"
        android:paddingTop="@dimen/tool_bar_top_padding"
        android:transitionName="actionBar" />

</RelativeLayout>
```

Создайте в папке layout файл drawer\_header.xml со следующим содержанием

```
<?xml version="1.0" encoding="utf-8"?>
<ImageView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:adjustViewBounds="true"
    android:orientation="vertical"
    android:scaleType="fitCenter"
    android:src="@drawable/header"></ImageView>
```

этот файл — разметка для верхней части Drawer'a, в которой находится картинка. Теперь положите в папку \res\drawable\ любую картинку с именем header.jpg, которая будет отображаться в верхней части Drawer'a, например эту:



Файл \res\strings.xml, содержащий строковые ресурсы, приведите к следующему виду

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">My Application</string>
    <string name="action_settings">Settings</string>
    <string name="drawer_item_home">Home</string>
    <string name="drawer_item_free_play">Free Play</string>
    <string name="drawer_item_custom">Custom</string>
    <string name="drawer_item_settings">Settings</string>
    <string name="drawer_item_help">Help</string>
    <string name="drawer_item_open_source">Open Source</string>
    <string name="drawer_item_contact">Contact</string>
</resources>
```

## Инициализация Navigation Drawer

В методе onCreate вашей MainActivity мы инициализируем Toolbar, добавьте после setContentView следующий код:

```
// Handle Toolbar
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```

Затем инициализируем и сам Navigation Drawer, добавьте ниже:

```
new Drawer()
    .withActivity(this)
    .withToolbar(toolbar)
    .withActionBarDrawerToggle(true)
    .withHeader(R.layout.drawer_header)
    .addDrawerItems(
        new PrimaryDrawerItem().withName(R.string.drawer_item_home).withIcon(FontAwesome.Icon.faw_home).withBadge("99").withIdentifier(1),
        new PrimaryDrawerItem().withName(R.string.drawer_item_free_play).withIcon(FontAwesome.Icon.faw_gamepad),
        new PrimaryDrawerItem().withName(R.string.drawer_item_custom).withIcon(FontAwesome.Icon.faw_eye).withBadge("6").withIdentifier(2),
        new SectionDrawerItem().withName(R.string.drawer_item_settings),
        new SecondaryDrawerItem().withName(R.string.drawer_item_help).withIcon(FontAwesome.Icon.faw_cog),
```

```

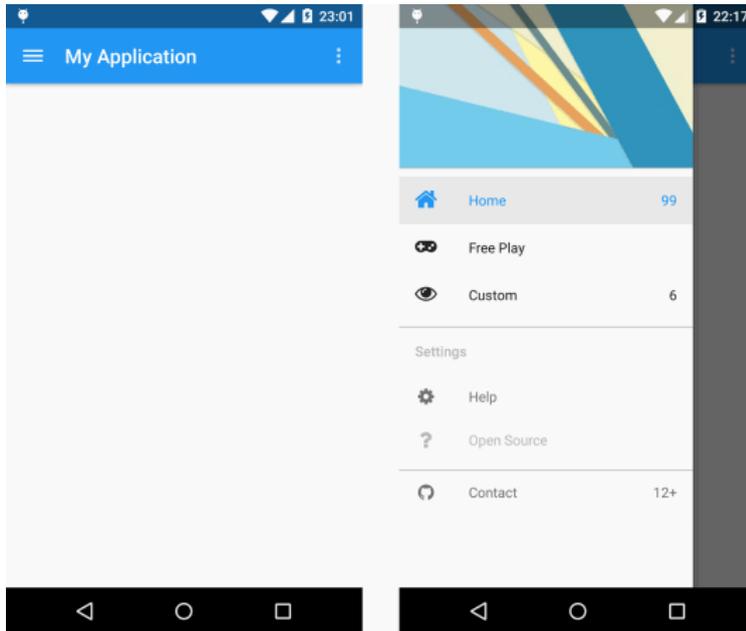
        new SecondaryDrawerItem().withName(R.string.drawer_item_open_source).withIcon(FontAwesome.Icon.faw_question).setEnabled(false),
        new DividerDrawerItem(),
        new SecondaryDrawerItem().withName(R.string.drawer_item_contact).withIcon(FontAwesome.Icon.faw_github).withBadge("12+").withIdentifier(1)
    )
    .build();

```

В случае появления ошибок, убедитесь, что ваша секция импортов в MainActivity выглядит так:

► [Секция импортов в MainActivity](#)

Теперь можно запустить приложение и оценить результат:



## Улучшения Navigation Drawer

Чтобы Navigation Drawer еще точнее соответствовал рекомендациям от Google, можно сделать следующие улучшения (см. полный листинг MainActivity в конце статьи):

- Скрывать клавиатуру при открытии NavigationDrawer:

```

.withOnDrawerListener(new Drawer.OnDrawerListener() {
    @Override
    public void onDrawerOpened(View drawerView) {
        InputMethodManager inputMethodManager = (InputMethodManager) MainActivity.this.getSystemService(Activity.INPUT_METHOD_SERVICE);
        inputMethodManager.hideSoftInputFromWindow(MainActivity.this.getCurrentFocus().getWindowToken(), 0);
    }
    @Override
    public void onDrawerClosed(View drawerView) {
    }
})

```

- Закрывать NavigationDrawer по нажатию системной кнопки «Назад»:

```

@Override
public void onBackPressed() {
    if (drawerResult.isDrawerOpen()) {
        drawerResult.closeDrawer();
    }
    else {
        super.onBackPressed();
    }
}

```

- Обработать события клика и длинного клика на элементы Drawer'a
- Уменьшать/увеличивать значения бейджей

Реализацию всех этих улучшений вы можете посмотреть в полном листинге MainActivity:

► [Полный код MainActivity](#)

## Альтернативы

Хотелось бы обратить ваше внимание на то, что никакая библиотека не покрывает всех нужд всех программистов. Поэтому приведу список аналогичных библиотек, с подключением которых можно поиграться:

<https://github.com/neokree/MaterialNavigationDrawer>  
<https://github.com/HeinrichReimer/material-drawer>  
<https://github.com/kanytu/android-material-drawer-template>  
<https://github.com/balysv/material-menu>  
<https://github.com/ikimuhendis/LDrawer>  
<https://github.com/Zlate87/material-navigation-drawer-example>

Что же касается описанной в статье библиотеки, автор легко идет на контакт и очень оперативно реагирует на замечания и feature-requests, с ним можно пообщаться, [создав issue](#), например.

Если же вам вообще не подходит ни одна библиотека, то вы всегда можете написать свою :)

## Ссылки

Готовый пример из статьи на Github: <https://github.com/tral/MaterialDrawerSample>;  
 Готовый пример с фрагментами: <https://github.com/tral/MaterialDrawerFragmentSample>;  
 Библиотека MaterialDrawer от Mike Penz: <https://github.com/mikepenz/MaterialDrawer>  
 Google Material Design Guidelines: Navigation Drawer: <http://www.google.com/design/spec/patterns/navigation-drawer.html>;  
 Dashboards: [https://developer.android.com/about/dashboards/index.html?utm\\_source=ausdroid.net](https://developer.android.com/about/dashboards/index.html?utm_source=ausdroid.net);  
 Support Library: <https://developer.android.com/tools/support-library/index.html>

android, google, tutorial, actionbar, navigation drawer, support library, appcompat



Не роскошь, а средство общения  
 Смартфон Micromax Bolt A79

2490 рублей

МЕГАФОН



## Комментарии (21)

 **Bringoff** 16 февраля 2015 в 23:28 # +4 ↑ ↓

Что за мода пошла в каждом tutorialе разжевывать моменты, непонятные только тем, кто вообще не связан с темой вопроса? Хотелось бы найти на хабре человека, Кострому надо каждый раз пошагово подсказывать, как создать новый проект.

 **nizulko** 17 февраля 2015 в 08:26 # [h](#) ↑ +6 ↑ ↓

*Статья для меня. Никогда не был в Костроме.*

 **Bringoff** 17 февраля 2015 в 08:41 (комментарий был изменён) # [h](#) ↑ 0 ↑ ↓

Ошибочка вышла. *Этот 79 Swype...*

 **tmnhy** 17 февраля 2015 в 00:42 # +5 ↑ ↓

Все туторы по Navigation Drawer (и не только) — «Ding in sich», элемент интерфейса ради интерфейса. Почему бы, не добавить конкретных примеров вместо заглушек? Как строить интерфейс фрагментами или активити, как раз для новичков полезно.

 **Mako** 17 февраля 2015 в 08:46 # 0 ↑ ↓

- 1) Зачем делать «кастомную» Navigation Drawer, если в той же Android Studio, которая уже версии 1.0 есть шаблон приложения с боковым меню?
- 2) Зачем Вы используете стороннюю библиотеку MaterialDrawer от Mike Penz?
- 3) Если я правильно помню, Navigation Drawer не должен закрывать ActionBar. В том же шаблонном приложении (да и во многих

туториалах) при открытии/закрытии меню всегда меняется title + иконка

 **DZVang** 17 февраля 2015 в 09:10 # [h](#) [↑](#) +1 [↑](#) [↓](#)

1-2) Затем, что в библиотеке уже все реализовано по последним гайдам и нужно просто добавить необходимые элементы для того чтобы всё было как нужно. В Android Studio же лежит неактуальный шаблон.  
3) Так было раньше, по новым гайдам Navigation Drawer перекрывает всё.

 **tralchonok** 17 февраля 2015 в 09:53 # [h](#) [↑](#) +2 [↑](#) [↓](#)

Да, вы верно на все ответили. Добавлю, что Google многих удивил своей непоследовательностью в подходах к работе с Navigation Drawer. Например, долго рекламировали значок, который красиво преобразуется из «гамбургера» в «стрелку назад». По последним же гайдлайнам, он скрывается за Drawer'ом (хотя, если приглядеться, вы увидите часть анимации).

 **DZVang** 17 февраля 2015 в 11:04 # [h](#) [↑](#) 0 [↑](#) [↓](#)

Согласен, перекрытие анимации иконки как то нелогично выглядит совсем.

 **Bringoff** 17 февраля 2015 в 12:05 # [h](#) [↑](#) 0 [↑](#) [↓](#)

Ну, это не большая проблема. У гугла, судя по их программам, какого-то определенного мнения нету. Со стандартным drawer-ом прокатывает `marginTop="?attr/actionBarSize"` — и перекрытия нету. Если так хочется показать анимацию.

 **DZVang** 17 февраля 2015 в 12:16 # [h](#) [↑](#) 0 [↑](#) [↓](#)

Понятно, что это решаемо, но привычка то осталась) Гугл вроде старается все к единому стилю привести, то есть у Gmail, Inbox, Keep, Play Market, Play Music Navigation Drawer везде перекрывает ActionBar.

 **Bringoff** 17 февраля 2015 в 12:19 # [h](#) [↑](#) 0 [↑](#) [↓](#)

А в Google Drive, Google Calendar, Фото не перекрывается. И поди их пойми) Google никогда строгостью к дизайну не отличался.

 **DZVang** 17 февраля 2015 в 12:28 # [h](#) [↑](#) 0 [↑](#) [↓](#)

Действительно, вы правы, я то уже подумал, что наконец гугл начал делать приложения в одном стиле, а вот нет.

 **tralchonok** 17 февраля 2015 в 13:41 # [h](#) [↑](#) 0 [↑](#) [↓](#)

на Nexus 5 Android 5.0.1 в Календаре и Диске перекрывается Action Bar, но не перекрывается Status Bar. В «Фото», да, даже Action Bar не перекрывается. Может быть просто не успели обновить приложение?

► [СКРИНЫ](#)

 **Bringoff** 17 февраля 2015 в 13:47 (комментарий был изменён) # [h](#) [↑](#) 0 [↑](#) [↓](#)

Странно, на 4.4 вот так:

► [СКРИНЫ](#)

 **tralchonok** 17 февраля 2015 в 14:01 # [h](#) [↑](#) 0 [↑](#) [↓](#)

Возможно, Google для данных приложений использует Multiple APK Support и мы видим просто разные арк... Так или иначе, на Lollipop сейчас только одно исключение из правила «перекрывать Action Bar» в гугловых сервисах, надеюсь, они его скоро обновят, чтоб **своим же гайдлайнам** соответствовать.

 **Bringoff** 17 февраля 2015 в 09:14 # 0 [↑](#) [↓](#)

В общем, попробовал эту либу. С какой радости она оверрайдит цвета, заданные в теме как primary, primaryDark и т. д? Оказалось, чтобы использовались мои цвета, надо называть их определенным образом. А если у меня несколько тем? В общем, либа написана... нехорошо.

 **tralchonok** 17 февраля 2015 в 10:12 # [h](#) [↑](#) +1 [↑](#) [↓](#)

вам просто нужно переопределить цвета в colors.xml приложения

► [напримр, вот так](#)

 **Bringoff** 17 февраля 2015 в 10:26 # [h](#) [↑](#) 0 [↑](#) [↓](#)

Нет, вы не поняли, то, что называть цвета нужно только так и не иначе, я понял. Но если мне надо иметь возможность создать несколько тем? К примеру, дневную, ночную в нескольких вариациях (с оранжевым, синим и зеленым toolbar-ом)? Так я могу сделать N тем, с parent=«BaseAppTheme» и у каждой задать colorPrimary, colorAccent и другие, и одной строчкой в коде между ними переключаться. А с этой либой так не пойдет, по-моему.

 **tralchonok** 17 февраля 2015 в 10:44 #   0  

хм, ну вы можете переехать темы либы темами вашего приложения. Я так понял, в либе темы исключительно как примеры для наглядности.

 **Bringoff** 17 февраля 2015 в 10:49 (комментарий был изменён) #   0  

Ну смотрите, у меня в тестовом приложении была одна тема:

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
    <item name="colorPrimary">@color/color_primary</item> //названы цвета не так, как в примере
    <item name="colorPrimaryDark">@color/color_primary_dark</item>
    <item name="colorAccent">@color/color_accent</item>
</style>
```

И для toolbar-а было задано `android:background="?attr/colorPrimary"`. При использовании обычного Navigation Drawer-а цвет берется с моей темы, но после того как в коде описанному в посте drawer-у присваивается параметр `.withToolBar(getToolBar())`, `colorPrimary` чудесным образом перенаправляется на голубой, захардкоренный в либе. Только после переименования всех цветов начало использовать мои цвета. Но если мне надо, скажем, 3 главных цвета и переключаться между ними, тут абсолютно непонятно, как действовать, ибо параметры, заданные в темах, по сути, игнорируются.

Я сильно не ковырялся, может, что-то с этим и можно сделать, но то, что каждый, пытающийся написать свой drawer, начинает городить огород, очень смущает.

 **tralchonok** 17 февраля 2015 в 10:56 #   0  

Я согласен с вами, по сути, сколько библиотек — столько и огородов. Кому-то подходит, кому-то нет.

В итоге, я дополню пост списком библиотек-аналогов и упомяну, что четкие парни пишут «под себя» :)

Только зарегистрированные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.