



Автоматизация темизации приложения. Генерация стилей, ресурсов и атрибутов.

Даниил Попов

Руководитель группы Android разработки

О чем будет доклад

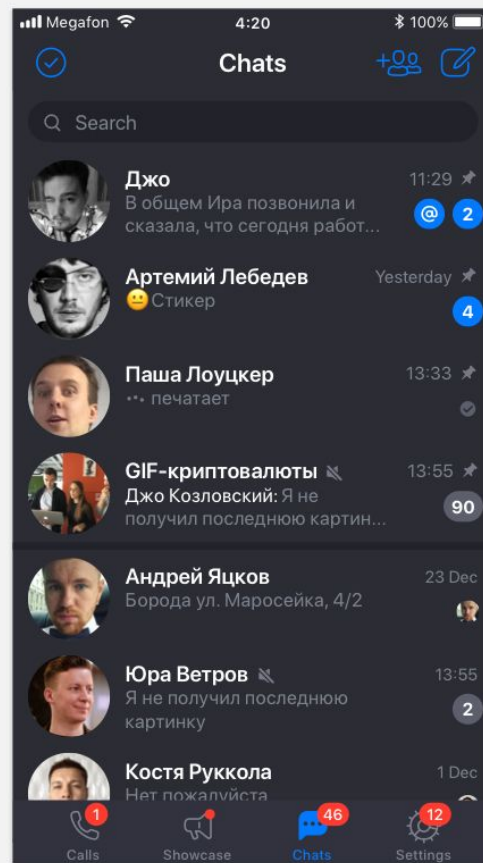
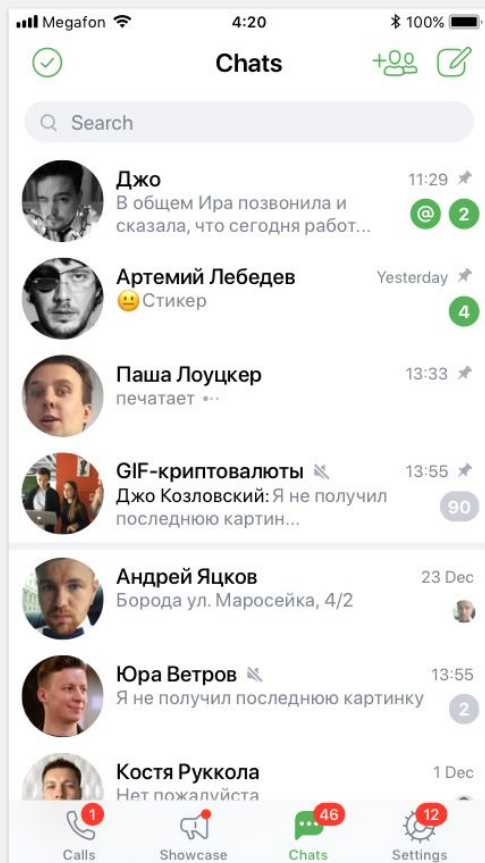
Что есть в докладе:

- Поддержка нескольких тем оформления
- Ошибки реализации
- Новый подход с автоматизацией
- Избавление от старого решения

Чего нет в докладе:

- Примеров кода генератора ресурсов

Как все начиналось

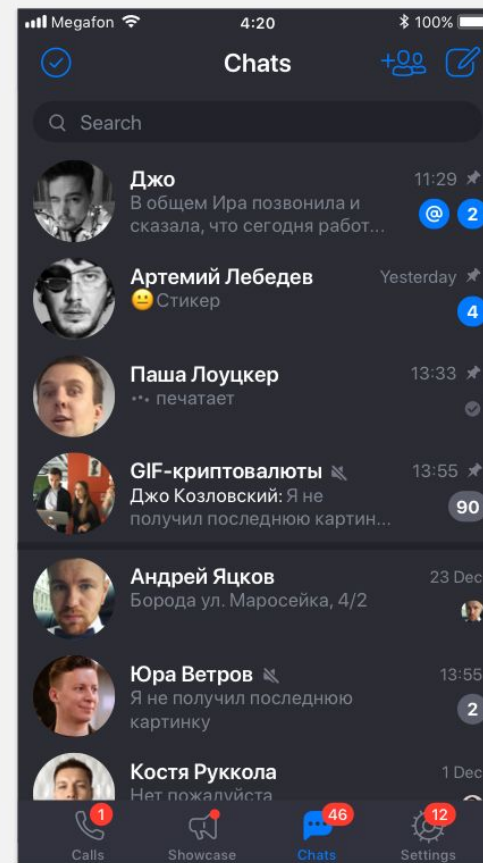
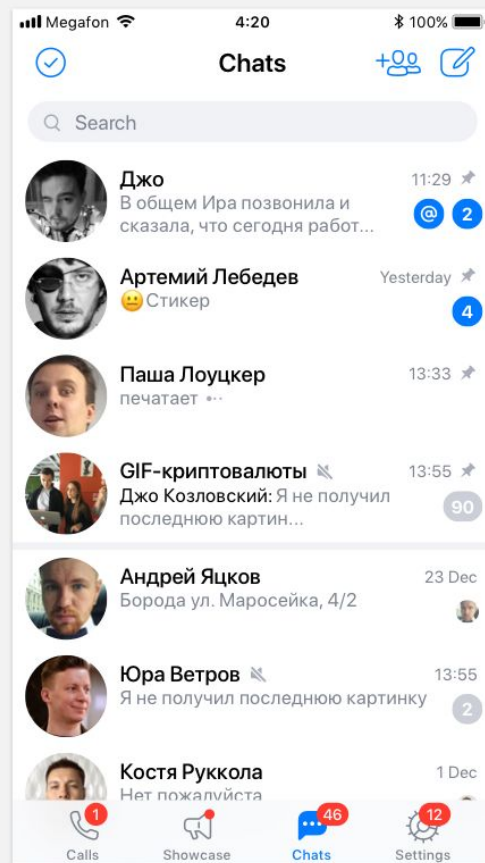
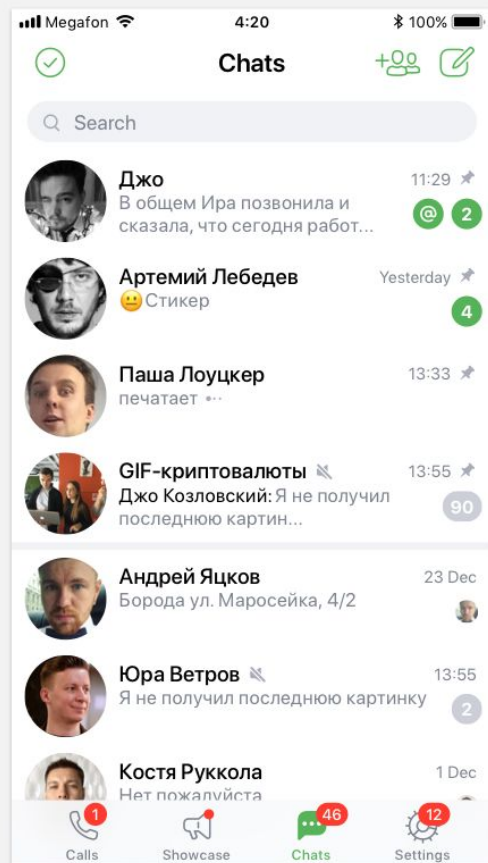


Разные темы. Ничего общего.

```
<style name="GreenTheme" parent="BaseTheme">
  <item name="colorAccent">@color/primary_green</item>
  <item name="chatStatusBarColor">@color/light_status_bar</item>
  <item name="rippleBg">@drawable/ripple_white</item>
  <item name="headerUnreadCounter">@drawable/header_unread_counter</item>
  ...
</style>

<style name="DarkTheme" parent="BaseTheme">
  <item name="colorAccent">@color/primary_dark</item>
  <item name="chatStatusBarColor">@color/dark_background</item>
  <item name="rippleBg">@drawable/ripple_black</item>
  <item name="headerUnreadCounter">@drawable/header_unread_counter_dark</item>
  ...
</style>
```

Двух тем должно хватить всем



Как мы это реализовали?

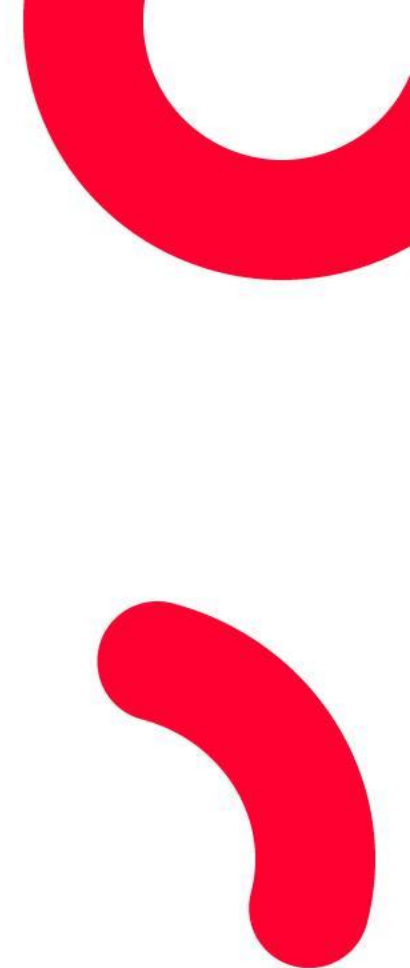
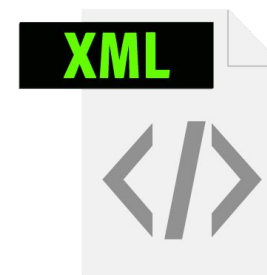
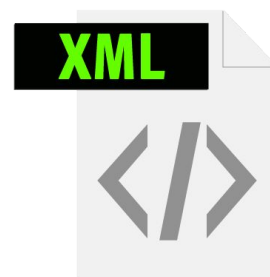
```
<style name="BlueTheme" parent="GreenTheme">  
  <item name="colorAccent">@color/primary_blue</item>  
</style>
```

К чему это привело и как это все поддерживать?

- Сложности с изменением ресурсов:
 - Где определен атрибут?
 - Где его менять?
 - Какие темы затронет?
- Сложности с добавлением ресурсов
 - Куда добавлять?
 - Какие темы затронет?

KitKat не только сладость, но и проблема

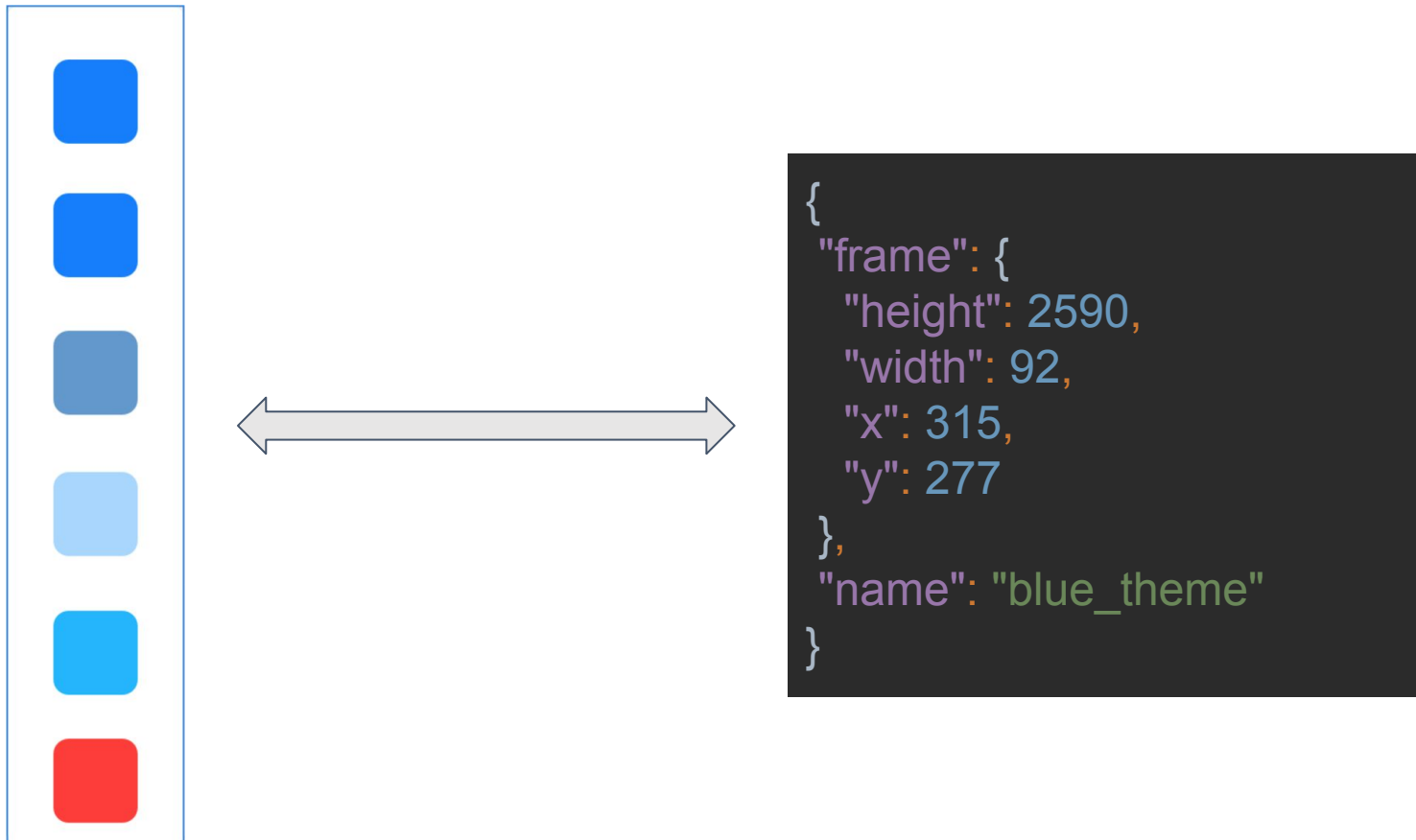
```
<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
  android:dither="true"  
  android:shape="rectangle">  
  <corners android:radius="4dp" />  
  <solid android:color="?colorPrimary" />  
</shape>
```



Палитра цветов

Тёмная тема	Зелёная тема	Синяя тема	
			#primary
			#primary_inverse
			#primary_pastel
			#primary_bright
			#secondary
			#secondary_attention

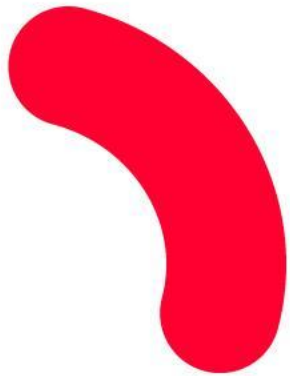
Магия парсинга Sketch файла. Извлекаем тему.



Магия парсинга Sketch файла. Извлекаем цвет.

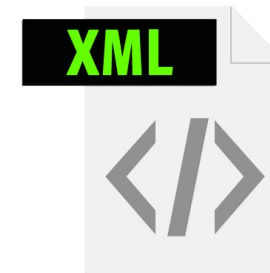
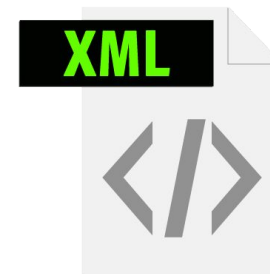


```
{
  "frame": {
    "height": 44, "width": 44, "x": 120, "y": 521
  },
  "name": "#primary_bright",
  "style": {
    "fills": [{
      "color": {
        "alpha": 1,
        "blue": 0.6627450980392157,
        "green": 0.4784313725490195,
        "red": 0.3254901960784314
      }
    }]
  }
}
```



Решение проблемы KitKat

```
<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
  android:dither="true"  
  android:shape="rectangle">  
  <corners android:radius="4dp" />  
  <solid android:color="?colorPrimary" />  
</shape>
```



Drawable ресурсы

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:dither="true"
    android:shape="rectangle">
    <corners android:radius="4dp" />
    <solid android:color="{COLOR.primary}" />
</shape>
```

XML



```
<solid android:color="@color/primary_green" />
```

XML



```
<solid android:color="@color/primary_blue" />
```

```
<solid android:color="@color/primary_dark" />
```

XML



Drawable в Drawable

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

  <item
    android:drawable="${DRAWABLE.button_primary_pressed}"
    android:state_pressed="true" />
  <item

    android:drawable="${DRAWABLE.button_primary_disabled}"
    android:state_enabled="false" />

  <item android:drawable="${DRAWABLE.button_primary_normal}" />

</selector>
```

Color State List ресурсы

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:color="${COLOR.primary_pastel}"
        android:state_pressed="true" />

    <item
        android:color="${COLOR.base_bright}"
        android:state_enabled="false" />

    <item android:color="${COLOR.primary}" />

</selector>
```

Как собирается тема

Для каждой темы:

1. Извлекаем цвета из палитры
2. Собираем xml-ресурсы
 - a. Drawable
 - b. ColorStateList

После этого:

1. Валидируем темы
2. Генерируем файлы с атрибутами, темами и ресурсами

Атрибуты темы

Цвета из палитры

#base_bright

?colorBaseBright

Drawable ресурсы

chat_header_bg.xml

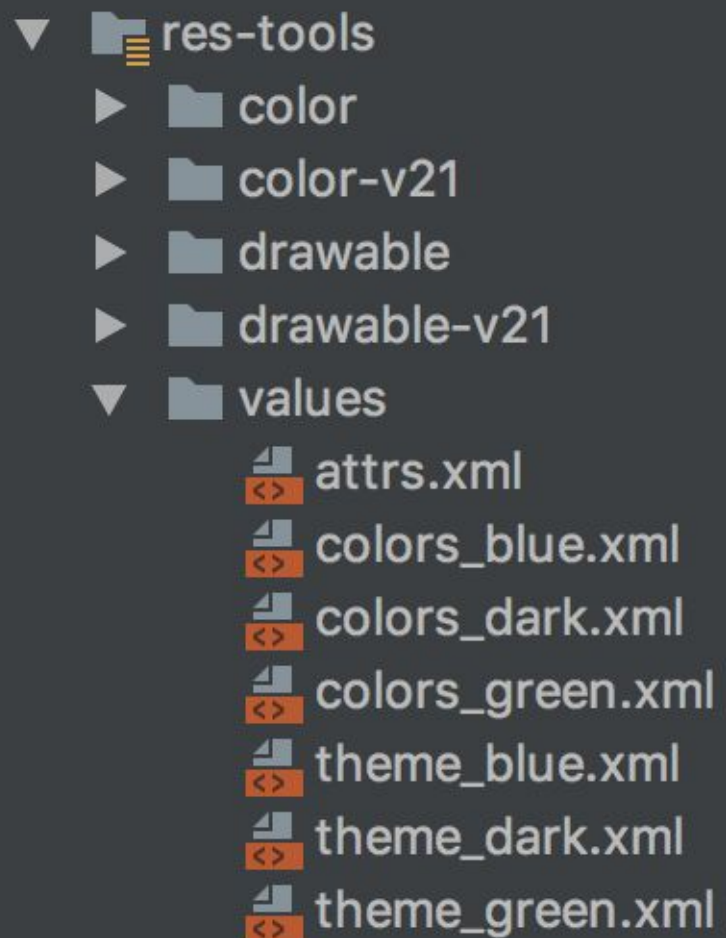
?drawableChatHeaderBg

Color-state-list ресурсы

button_primary_bg.xml

?colorButtonPrimaryBg

Что имеем на выходе



Атрибуты для всех собранных ресурсов

Палитра синей темы

Палитра темной темы

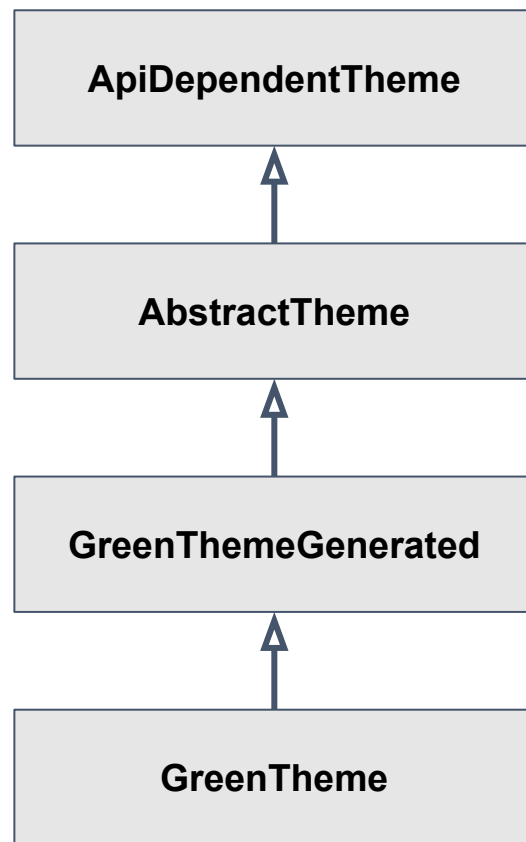
Палитра зеленой темы

\

- Отображение атрибутов на значения

/

Иерархия стилей



Версионирование

В системе контроля версий находятся:

- Sketch-файл с палитрой
- Шаблоны XML-ресурсов

Вне системы контроля версий:

- Все сгенерированные файлы
 - Атрибуты
 - Темы
 - Палитры
 - XML-ресурсы

Сборка проекта

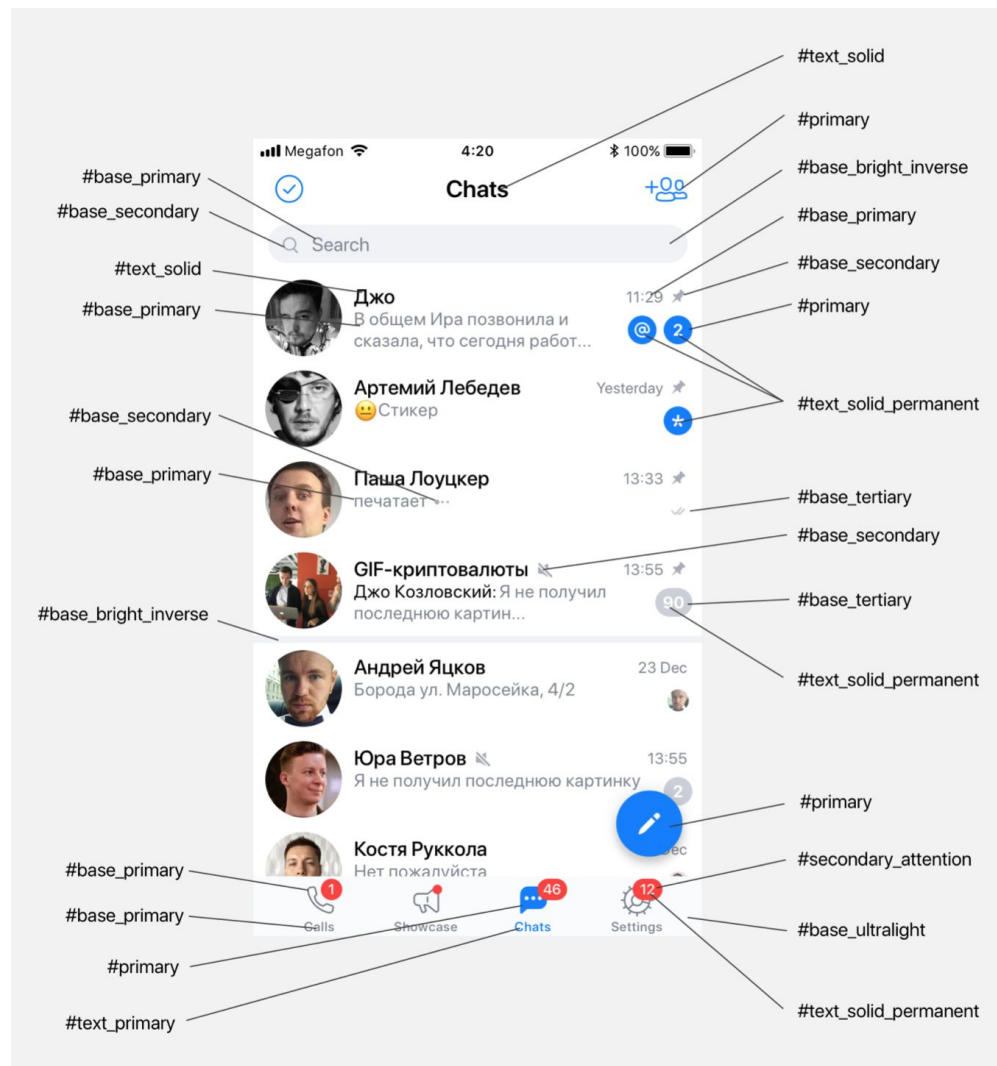
```
task generateResources(type: Exec) {  
    workingDir "${project.file('theme-resources/')}"  
    if (isWindows()) {  
        commandLine = ['cmd', '/C', 'start', 'generate.bat']  
    } else {  
        commandLine './generate.sh'  
    }  
}  
  
...  
android.applicationVariants.all { variant ->  
    variant.mergeResources.dependsOn tasks.generateResources  
}  
  
...  
main.res.srcDirs += "${project.buildDir}/generated/res-tools"
```

Генерация ресурсов

*Встраиваемся в
процесс обработки
ресурсов*

*Указываем путь до
результата генерации*

Как это выглядит в макетах

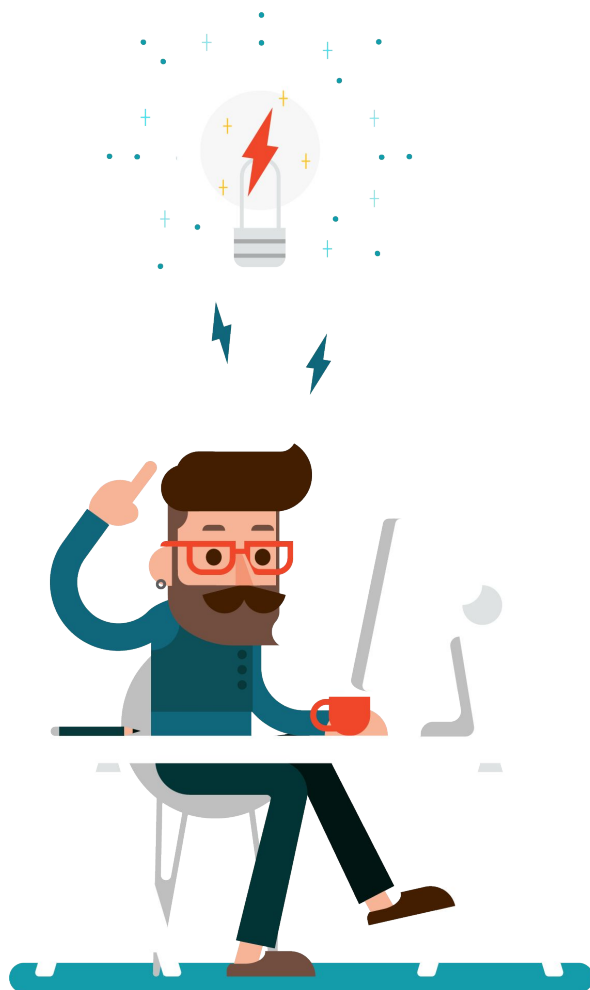


Как это выглядит в разметке

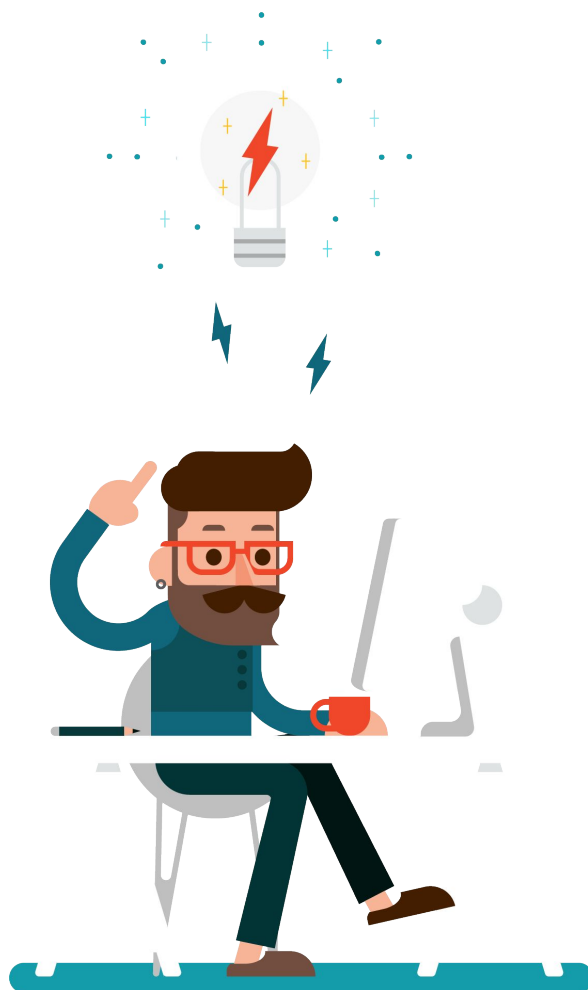
```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    ...
    android:textColor="?colorTextSolidPermanent"/>

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    ...
    android:background="?drawableBottomBarBg">
```

Разделение ответственности



Разделение ответственности



Дизайнер



Разработчик

Как избавиться от старых цветов

1. Переделать все разом
 - + Нет legacy ресурсов
 - + Приложение выглядит консистентно
 - Требуется значительных совместных усилий разработчика и дизайнера
2. Переходить постепенно
 - + Не требует больших единовременных временных затрат
 - В коде остаются deprecated ресурсы
 - Разные экраны приложения могут выглядеть неконсистентно

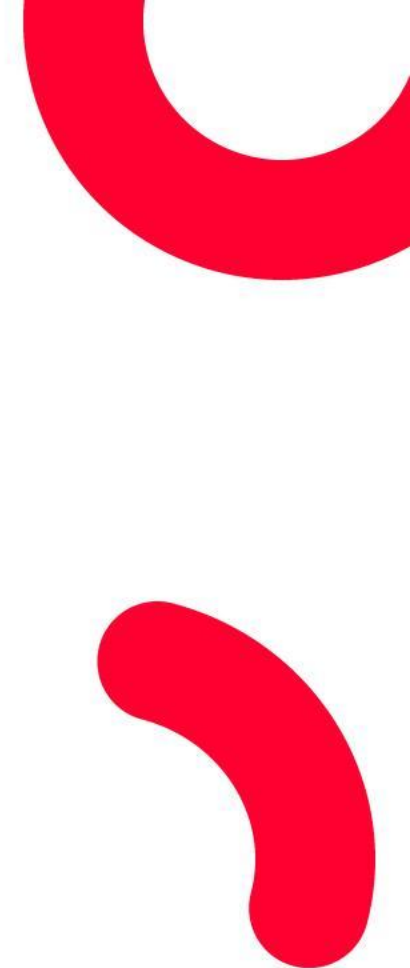
Profit!!!

1. Простота поддержки ресурсов
2. Легкость работы с темами оформления
3. Разделение ответственности
4. Ограниченная палитра цветов
5. Упрощение слияния изменений
6. Кроссплатформенность

Всегда есть, что улучшить

- Темизация таких ресурсов как:
 - 9-patch
 - Обычные картинки
- Не производить генерацию ресурсов когда нет изменений
- Опубликовать решение в open source

Ждем вас в нашей команде!



СПАСИБО!

