

# Widgets

# android



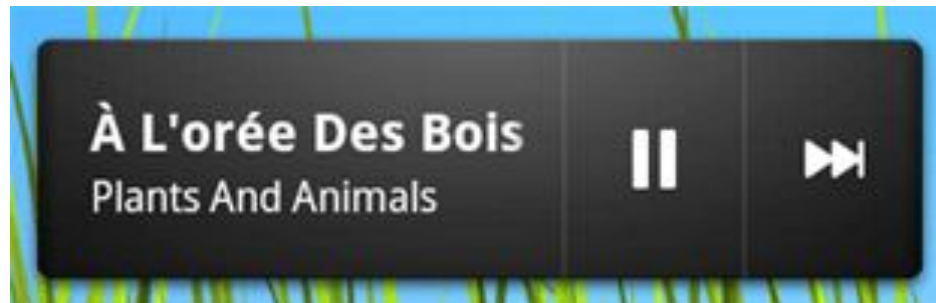
77°

Mountain View  
Sunny  
79° 57°



# Что такое виджет?

App Widgets are miniature application views that can be embedded in other applications and receive periodic updates. These views are referred to as Widgets in the user interface, and you can publish one with an App Widget provider. An application component that is able to hold other App Widgets is called an App Widget host. The screenshot below shows the Music App Widget:



# ТИПЫ ВИДЖЕТОВ

As you begin planning your widget, think about what kind of widget you're trying to build. Widgets typically fall into one of the following categories:

- Information widgets
- Collection widgets
- Control widgets
- Hybrid widgets

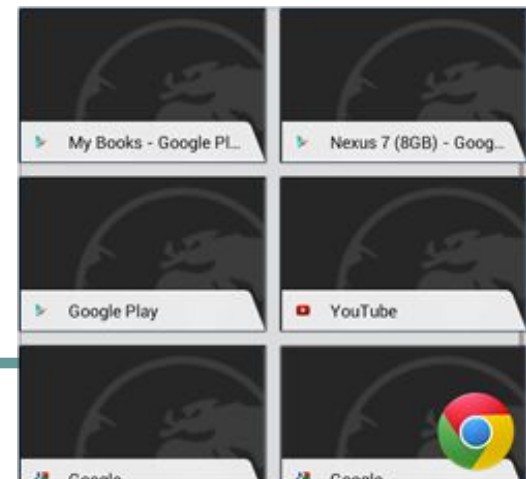
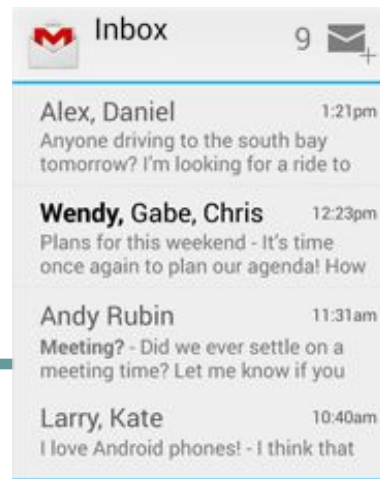
# Информационные виджеты

Information widgets typically display a few crucial information elements that are important to a user and track how that information changes over time. Good examples for information widgets are weather widgets, clock widgets or sports score trackers. Touching information widgets typically launches the associated app and opens a detail view of the widget information.



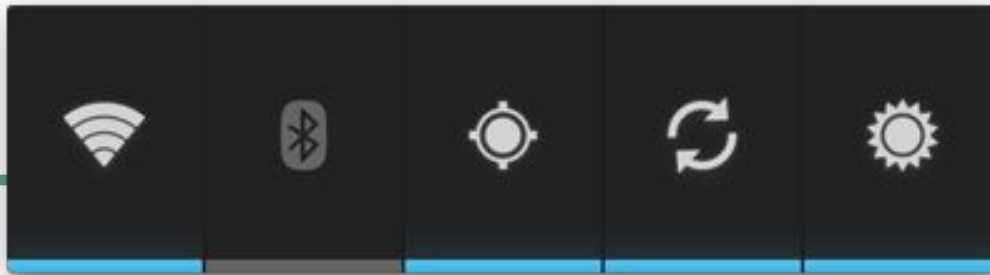
# Виджеты для коллекций

As the name implies, collection widgets specialize in displaying multitude elements of the same type, such as a collection of pictures from a gallery app, a collection of articles from a news app or a collection of emails/messages from a communication app. Collection widgets typically focus on two use cases: browsing the collection, and opening an element of the collection to its detail view for consumption. Collection widgets can scroll vertically.



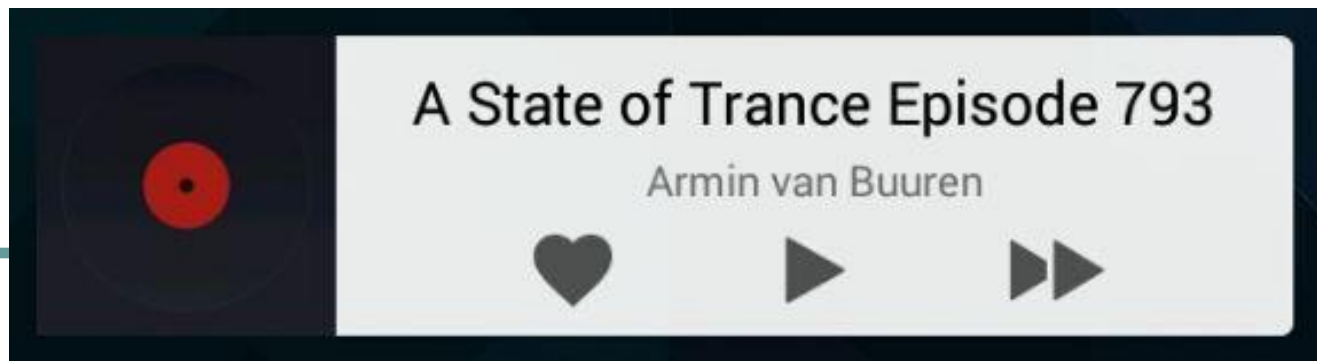
# Управляющие виджеты

The main purpose of a control widget is to display often used functions that the user can trigger right from the home screen without having to open the app first. Think of them as remote controls for an app. A typical example of control widgets are music app widgets that allow the user to play, pause or skip music tracks from outside the actual music app. Interacting with control widgets may or may not progress to an associated detail view depending on if the control widget's function generated a data set, such as in the case of a search widget.



# Виджеты смешанного типа

While all widgets tend to gravitate towards one of the three types described above, many widgets in reality are hybrids that combine elements of different types. For the purpose of your widget planning, center your widget around one of the base types and add elements of other types if needed. A music player widget is primarily a control widget, but also keeps the user informed about what track is currently playing. It essentially combines a control widget with elements of an information widget type.



# Ограничения по жестам

Because widgets live on the home screen, they have to co-exist with the navigation that is established there. This limits the gesture support that is available in a widget compared to a full-screen app. The only gestures available for widgets are:

- Touch
- Vertical swipe



# Содержимое виджета

Widgets are a great mechanism to attract a user to your app by "advertising" new and interesting content that is available for consumption in your app. Just like the teasers on the front page of a newspaper, widgets should consolidate and concentrate an app's information and then provide a connection to richer detail within the app; or in other words: the widget is the information "snack" while the app is the "meal." As a bottom line, always make sure that your app shows more detail about an information item than what the widget already displays.

# ОСНОВНЫЕ КЛАССЫ

To create an App Widget, you need the following:

- AppWidgetProviderInfo object - describes the metadata for an App Widget, such as the App Widget's layout, update frequency, and the AppWidgetProvider class. This should be defined in XML
- AppWidgetProvider class implementation - defines the basic methods that allow you to programmatically interface with the App Widget, based on broadcast events. Through it, you will receive broadcasts when the App Widget is updated, enabled, disabled and deleted
- View layout - defines the initial layout for the App Widget, defined in XML

# Объявление виджета

First, declare the [AppWidgetProvider](#) class in your application's AndroidManifest.xml file:

```
<receiver android:name="ExampleAppWidgetProvider" >
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>
    <meta-data android:name="android.appwidget.provider"
        android:resource="@xml/example_appwidget_info" />
</receiver>
```

The <receiver> element requires the **android:name** attribute, which specifies the [AppWidgetProvider](#) used by the App Widget.

The <meta-data> element specifies the [AppWidgetProviderInfo](#) resource and requires the following attributes:

- **android:name** - specifies the metadata name. Use **android.appwidget.provider** to identify the data as the [AppWidgetProviderInfo](#) descriptor.
- **android:resource** - specifies the [AppWidgetProviderInfo](#) resource location

# AppWidgetProviderInfo

The [AppWidgetProviderInfo](#) defines the essential qualities of an App Widget, such as its minimum layout dimensions, its initial layout resource, or how often to update the App Widget. Define the AppWidgetProviderInfo object in an XML resource using a single **<appwidget-provider>** element and save it in the project's **res/xml/** folder:

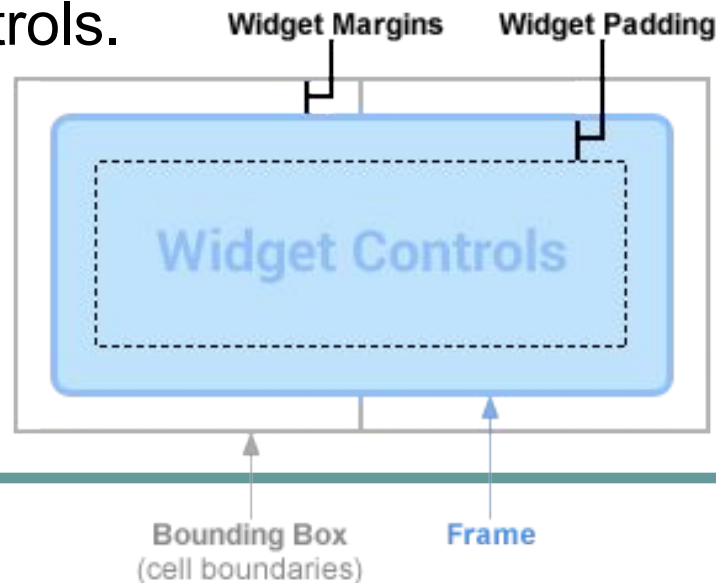
```
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="40dp"
    android:minHeight="40dp"
    android:updatePeriodMillis="86400000"
    android:previewImage="@drawable/preview"
    android:initialLayout="@layout/example_appwidget"
    android:configure="com.example.android.ExampleAppWidgetConfigure"
    android:resizeMode="horizontal|vertical"
    android:widgetCategory="home_screen">
</appwidget-provider>
```

# Атрибуты AppWidgetProvider

- The values for the **minWidth** and **minHeight** attributes specify the minimum amount of space the App Widget consumes by default. The default Home screen positions App Widgets in its window based on a grid of cells that have a defined height and width. If the values for an App Widget's minimum width or height don't match the dimensions of the cells, then the App Widget dimensions **round up** to the nearest cell size.

# Анатомия виджета

Typical Android app widgets have three main components: a bounding box, a frame, and the widget's graphical controls and other elements. App widgets can contain a subset of the View widgets in Android; supported controls include text labels, buttons, and images. Well-designed widgets leave some margins between the edges of the bounding box and the frame, and padding between the inner edges of the frame and the widget's controls.

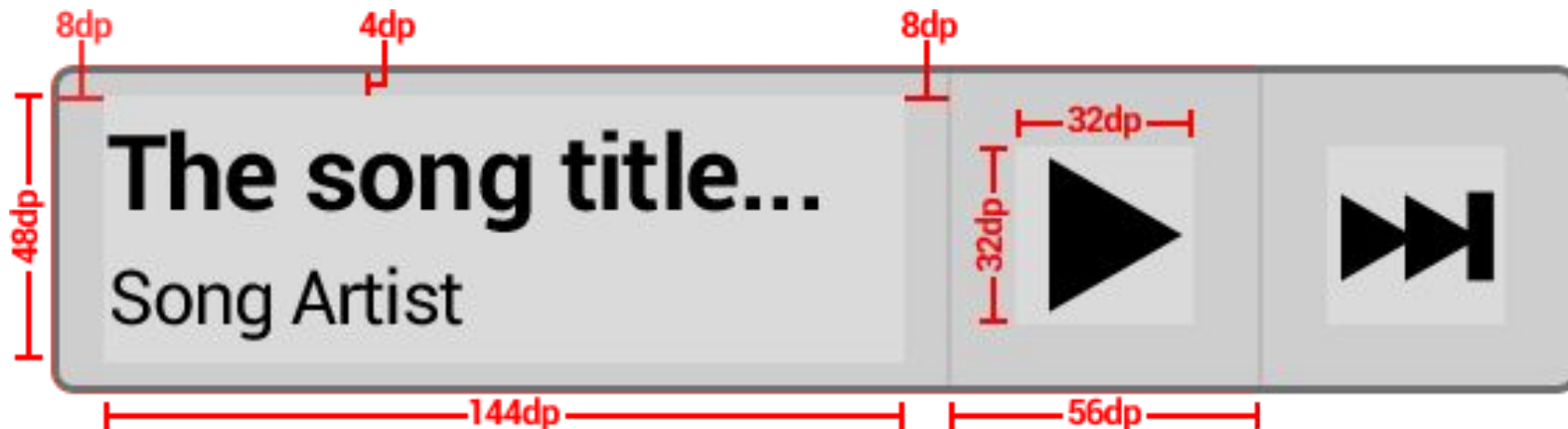


# Определение размера виджета

When users add a widget to their home screen, it will generally occupy more than the minimum width and height you specify. Home screens offer users a grid of available spaces into which they can place widgets and icons. This grid can vary by device; for example, many handsets offer a 4x4 grid, and tablets can offer a larger, 8x7 grid. **When your widget is added, it will be stretched to occupy the minimum number of cells, horizontally and vertically, required to satisfy its `minWidth` and `minHeight` constraints.** While the width and height of a cell — as well as the amount of automatic margins applied to widgets — may vary across devices, you can use the table below to roughly estimate your widget's minimum dimensions, given the desired number of occupied grid cells:

# of Cells (Columns or Rows)	Available Size (dp) ( <code>minWidth</code> or <code>minHeight</code> )
1	40dp
2	110dp
3	180dp
4	250dp
...	...
$n$	$70 \times n - 30$

# Определение размера виджета



Example calculations are below:

- **minWidth** =  $144dp + (2 \times 8dp) + (2 \times 56dp) = \mathbf{272dp}$
- **minHeight** =  $48dp + (2 \times 4dp) = \mathbf{56dp}$

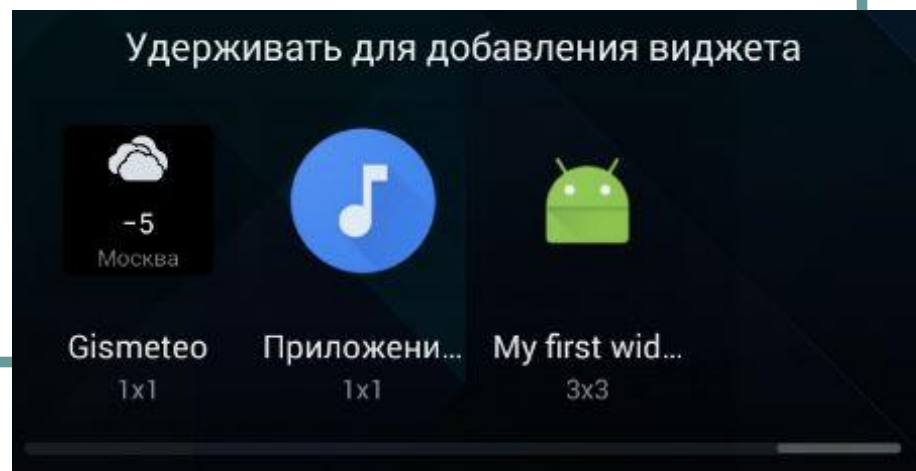
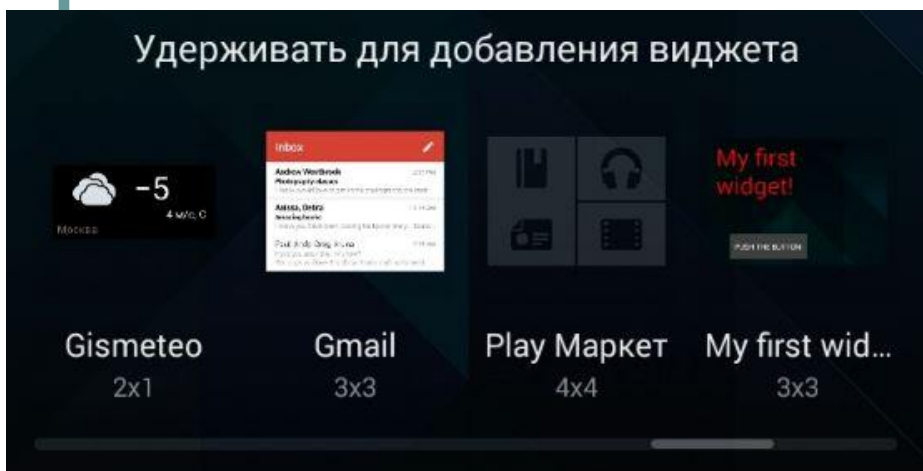


# Атрибуты AppWidgetProvider

- The **updatePeriodMillis** attribute defines how often the App Widget framework should request an update from the AppWidgetProvider attribute defines how often the App Widget framework should request an update from the AppWidgetProvider by calling the onUpdate() callback method. The actual update is not guaranteed to occur exactly on time with this value and we suggest updating as infrequently as possible — perhaps no more than once an hour to conserve the battery. If the device is asleep when it is time for an update (as defined by updatePeriodMillis) then the device

# Атрибуты AppWidgetProvider

- The **initialLayout** attribute points to the layout resource that defines the App Widget layout
- The **previewImage** attribute specifies a preview of what the app widget will look like after it's configured, which the user sees when selecting the app widget. If not supplied, the user instead sees your application's launcher icon:

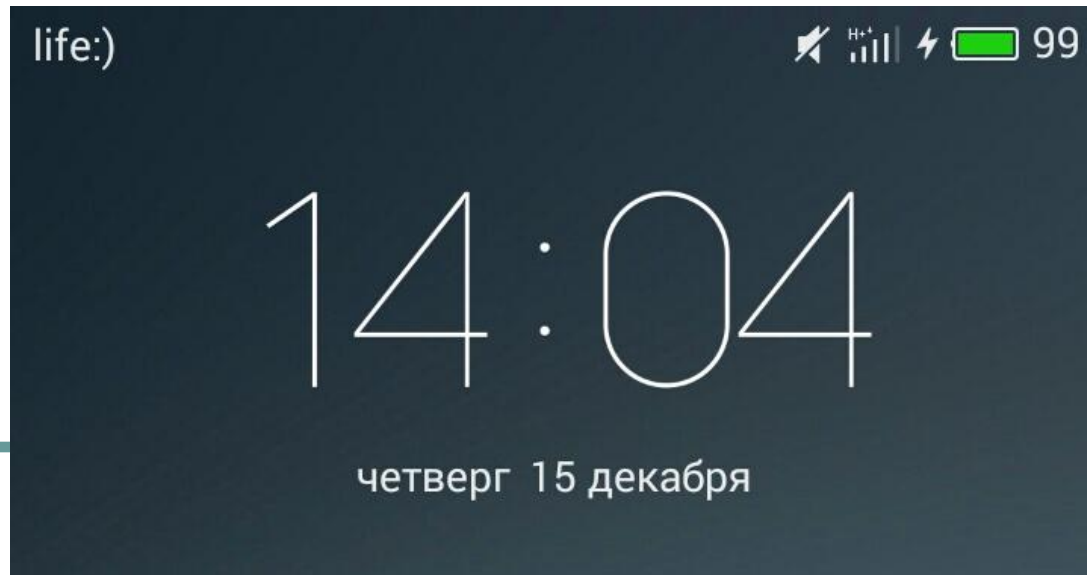


# Атрибуты AppWidgetProvider

- The **resizeMode** attribute specifies the rules by which a widget can be resized. You use this attribute to make homescreen widgets resizeable — horizontally, vertically, or on both axes. Users touch-hold a widget to show its resize handles, then drag the horizontal and/or vertical handles to change the size on the layout grid. Values for the resizeMode attribute include "horizontal", "vertical", and "none". To declare a widget as resizeable horizontally and vertically, supply the value "horizontal|vertical"

# Атрибуты AppWidgetProvider

- The **widgetCategory** attribute declares whether your App Widget can be displayed on the home screen (`home_screen`), the lock screen (`keyguard`), or both. Only Android versions lower than 5.0 support lock-screen widgets! For Android 5.0 and higher, only `home_screen` is valid...

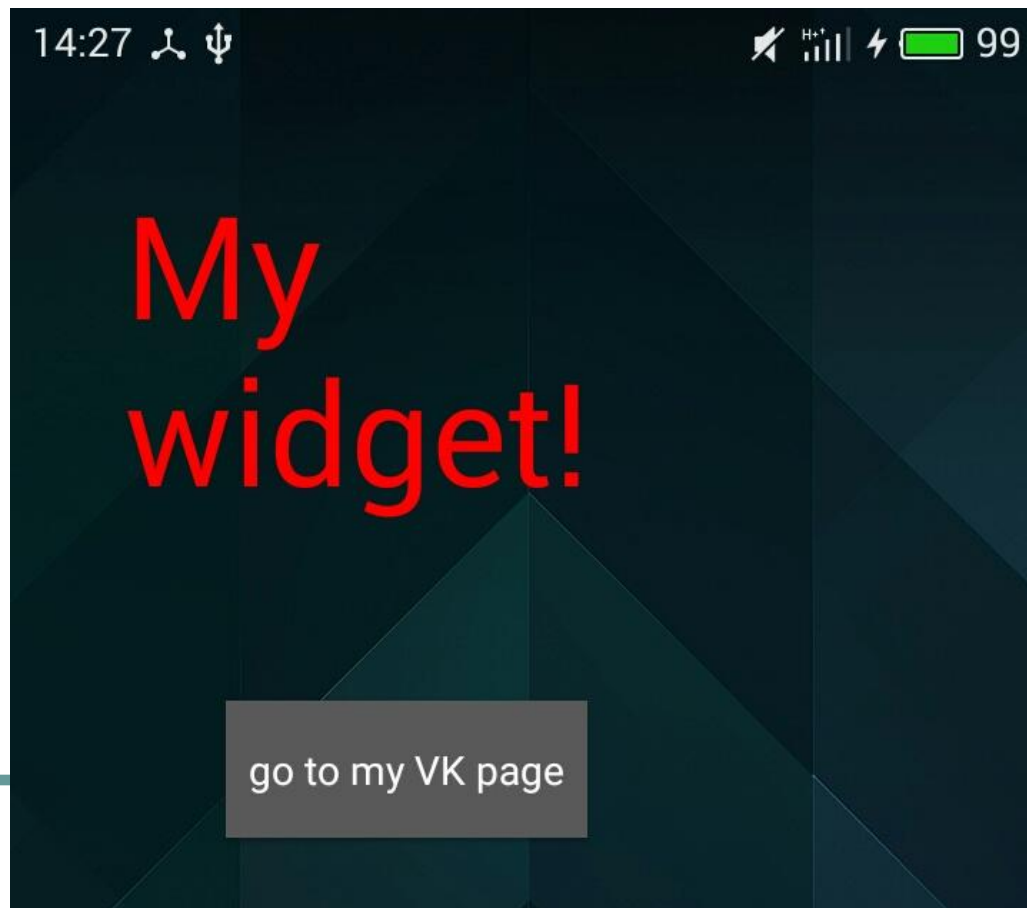


# Создание макета виджета

You must define an initial layout for your App Widget in XML and save it in the project's **res/layout/** directory. You can design your App Widget using the View objects listed below. Creating the App Widget layout is simple if you're familiar with Layouts directory. You can design your App Widget using the View objects listed below. Creating the App Widget layout is simple if you're familiar with Layouts. However, you must be aware that App Widget layouts are based on RemoteViews, which do not support every kind of layout or view widget. App Widget can support the following **layout classes**: FrameLayout: FrameLayout, LinearLayout: FrameLayout, LinearLayout, RelativeLayout: FrameLayout, LinearLayout, RelativeLayout and GridLayout. And the following **widget classes**: AnalogClock: AnalogClock, Button: AnalogClock, Button, ImageButton: AnalogClock, Button, ImageButton, ImageView: AnalogClock, Button, ImageButton, ImageView, ProgressBar: AnalogClock, Button, ImageButton, ImageView, TextView: AnalogClock, Button, ImageButton, ImageView,

# Пример виджета

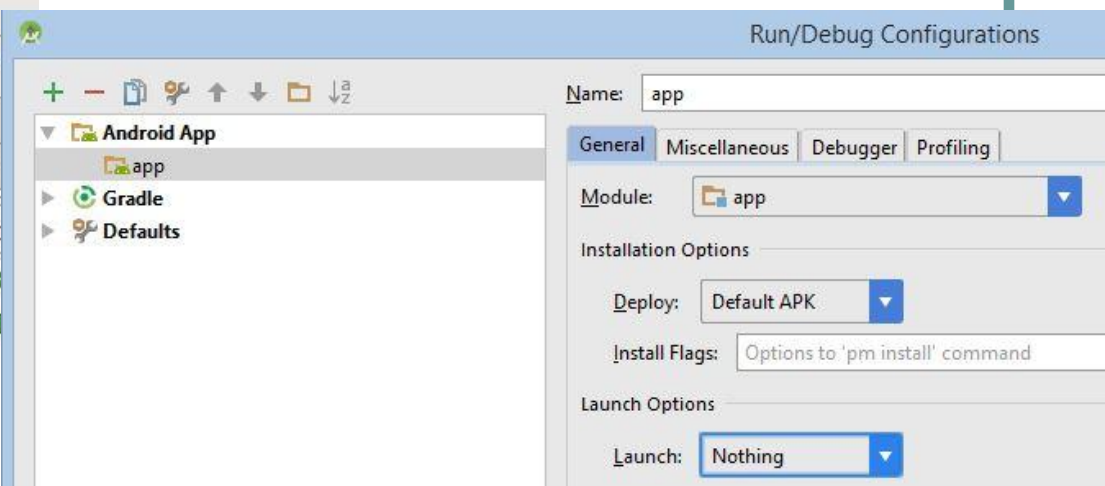
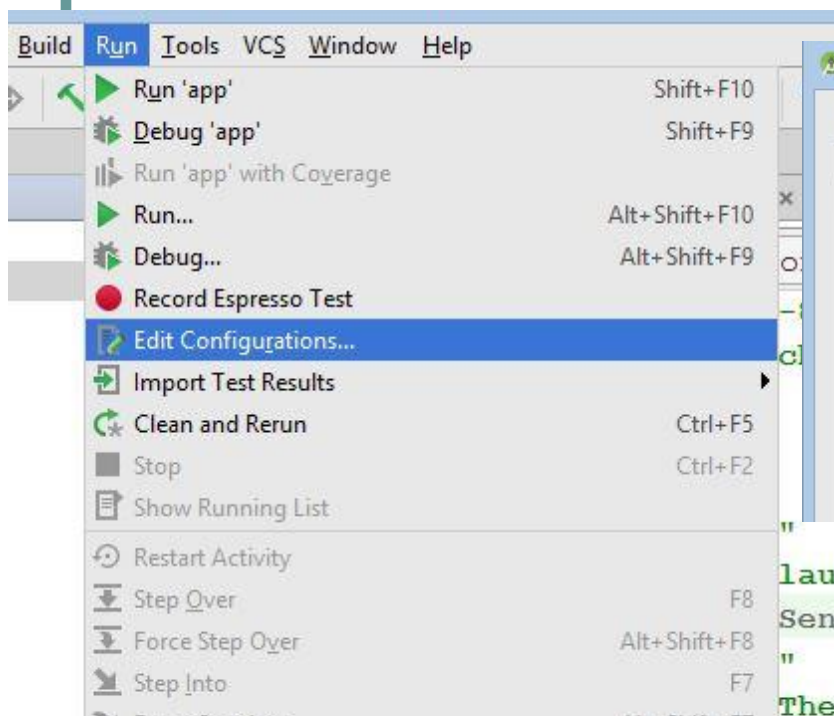
<https://git.io/v1Mir>



# Проблемы при запуске

Если при сборке студия выводит сообщение вроде **Error running app: Default Activity not found**, тогда поможет:

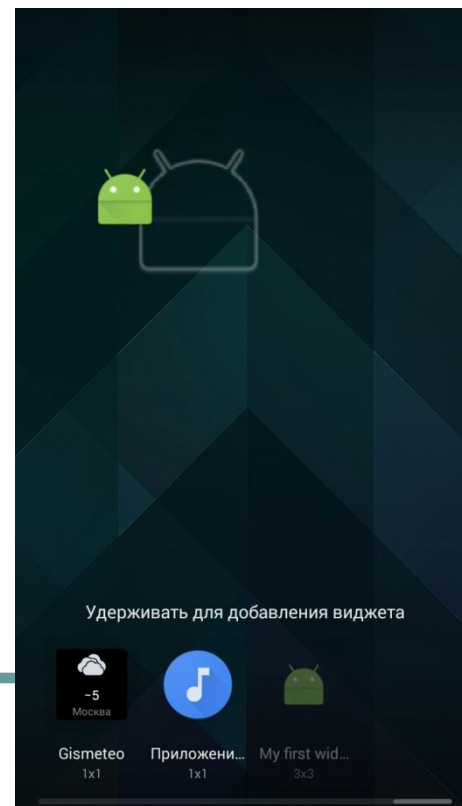
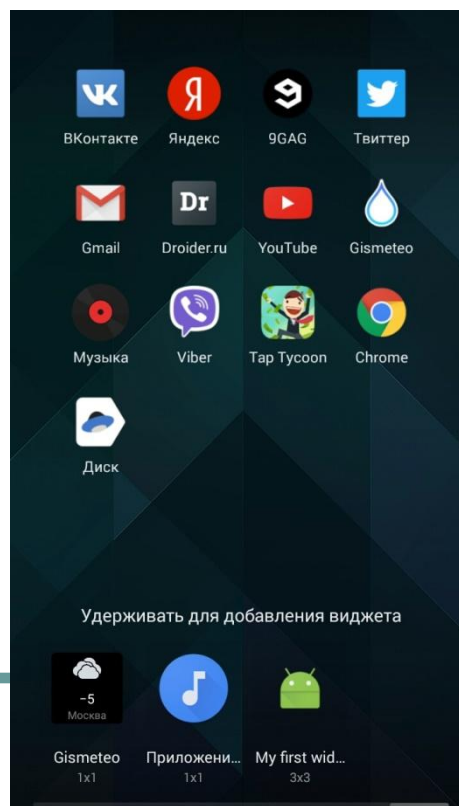
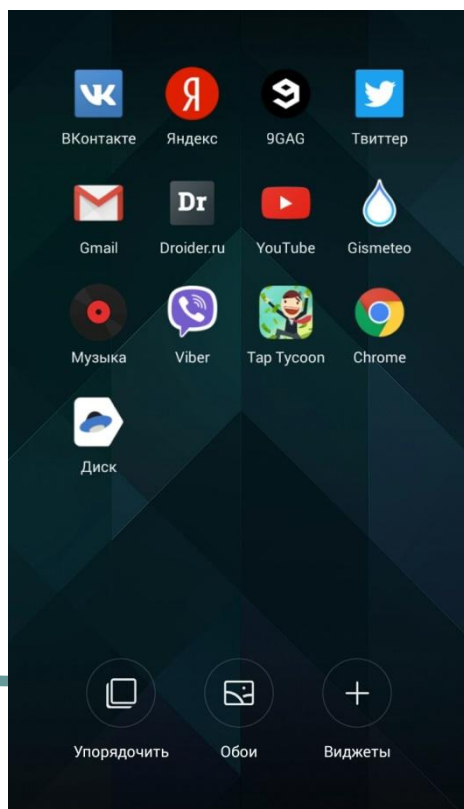
- File > Invalidate caches / Restart...
- Build > Rebuild project
- Run > Edit configurations > Android App > app > Launch Options > Launch: Nothing



<http://stackoverflow.com/questions/40134282/widget-only-app-default-activity-not-found>

# Установка виджета

On most Android devices, you'll start out by long-pressing a blank, available space on your home screen — not on an icon or the app launcher. Just hold your finger down on the screen.





# Обновление виджета

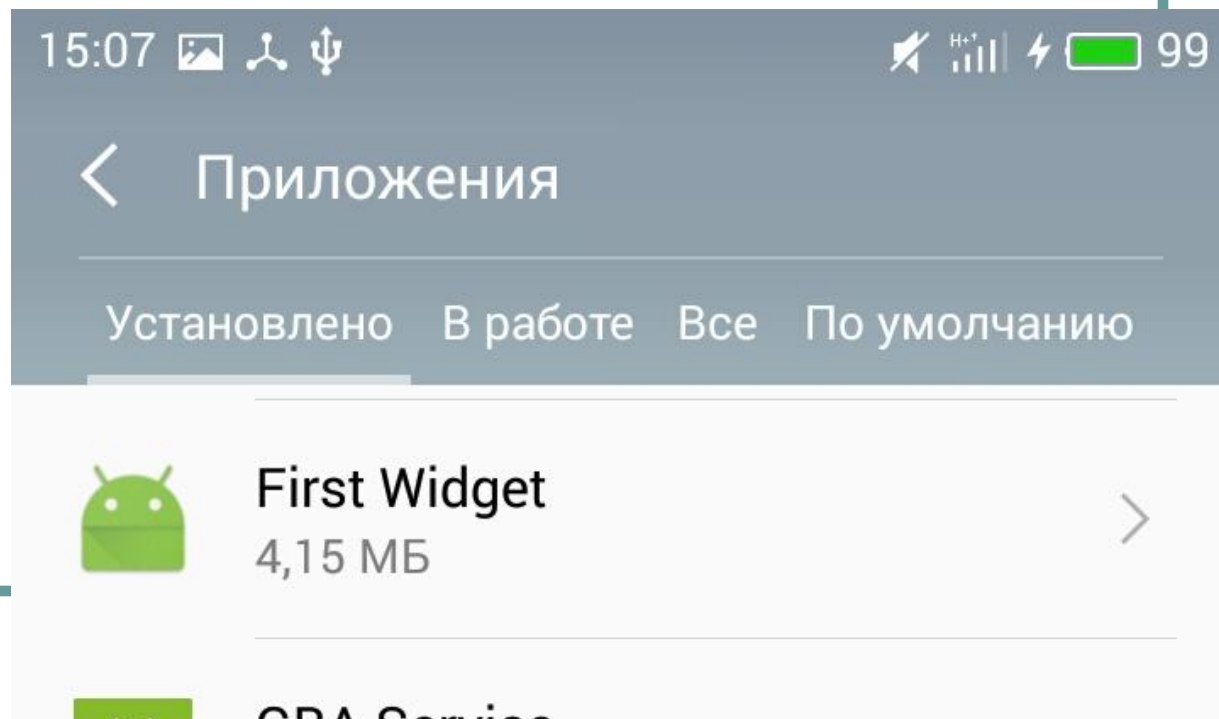
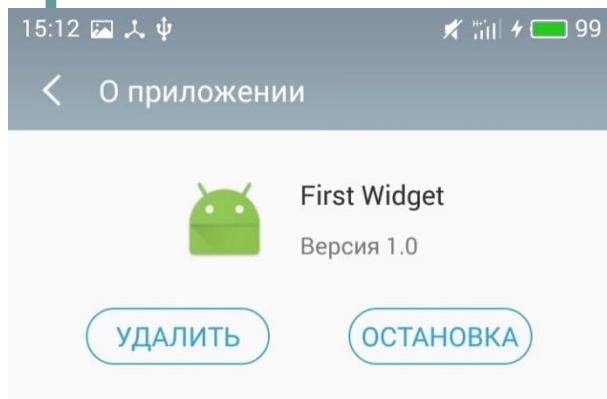
## <https://git.io/v1Slw>

- Особенность параметра **updatePeriodMillis**:  
<https://developer.android.com/reference/android/appwidget/AppWidgetProviderInfo.html#updatePeriodMillis> – минимально допустимое значение 1800000 мс (30 минут), впрочем AlarmManager или сервис позволит решить эту проблему:  
<http://stackoverflow.com/questions/3310264/frequently-updating-widgets-more-frequently-than-what-updateperiodmillis-allows>
- Добавлена реакция на создание и удаление виджета

# Как удалить виджет из системы

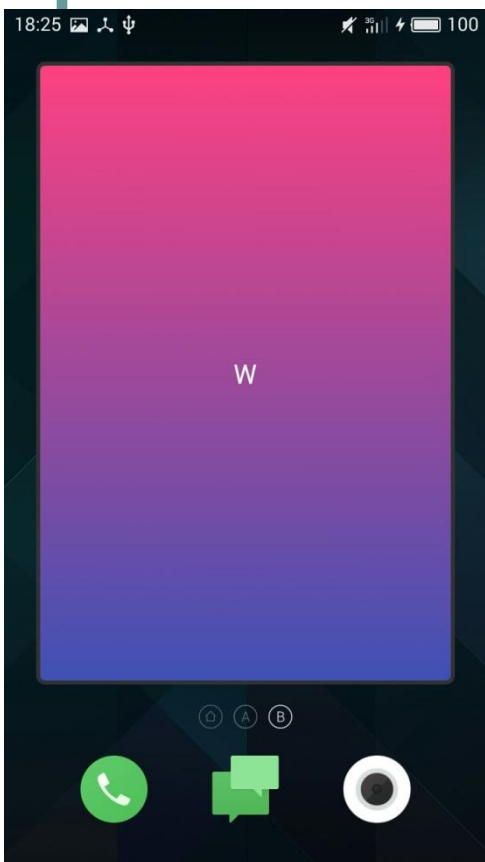
- Чтобы удалить виджет с рабочего стола, удерживайте палец на окне виджета в течении 1-2 секунд и переместите виджет в корзину
- Чтобы удалить виджет из системы, зайдите в Настройки > Приложения > Установленные приложения, выберите из списка нужный виджет и в его меню выберите "Удалить"

<http://allreviews.com.ua/faq/132-kak-ustanovit-i-kak-udalit-vidzheta-na-android.html>

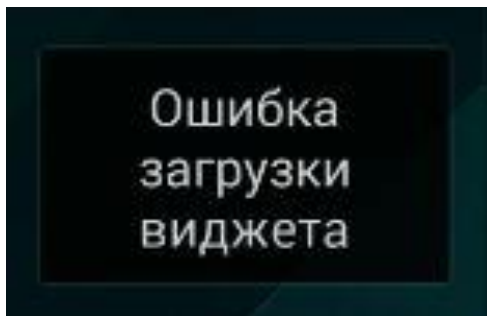


# Виджет – понятие растяжимое

<https://git.io/v1S6T>



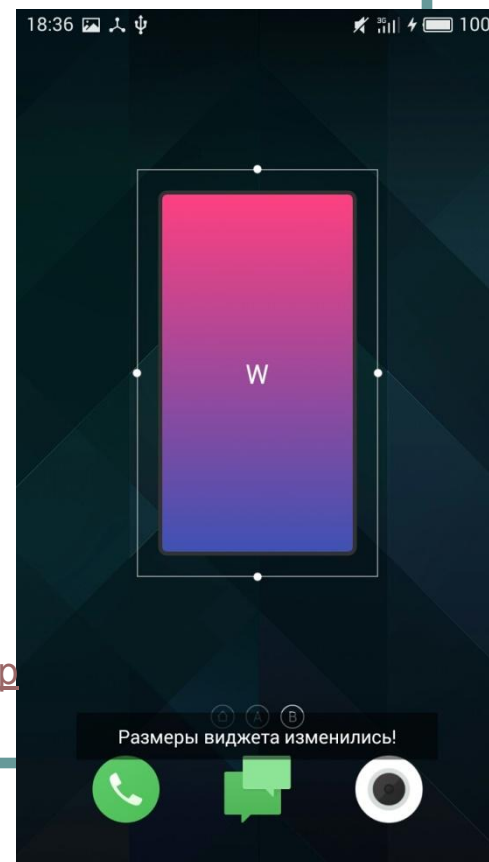
Если вдруг в процессе тестирования виджета на рабочем столе появится что-то вроде тогда поможет следующий набор инструкций:



**Build > Clean Project**, затем **Build > Make Project**, опционально **Rebuild Project**, и наконец **Run > Run app...**

<http://stackoverflow.com/questions/11792071/xml-shape-e-gradient-doesnt-show-on-my-widget>

<http://angrytools.com/android/button/>



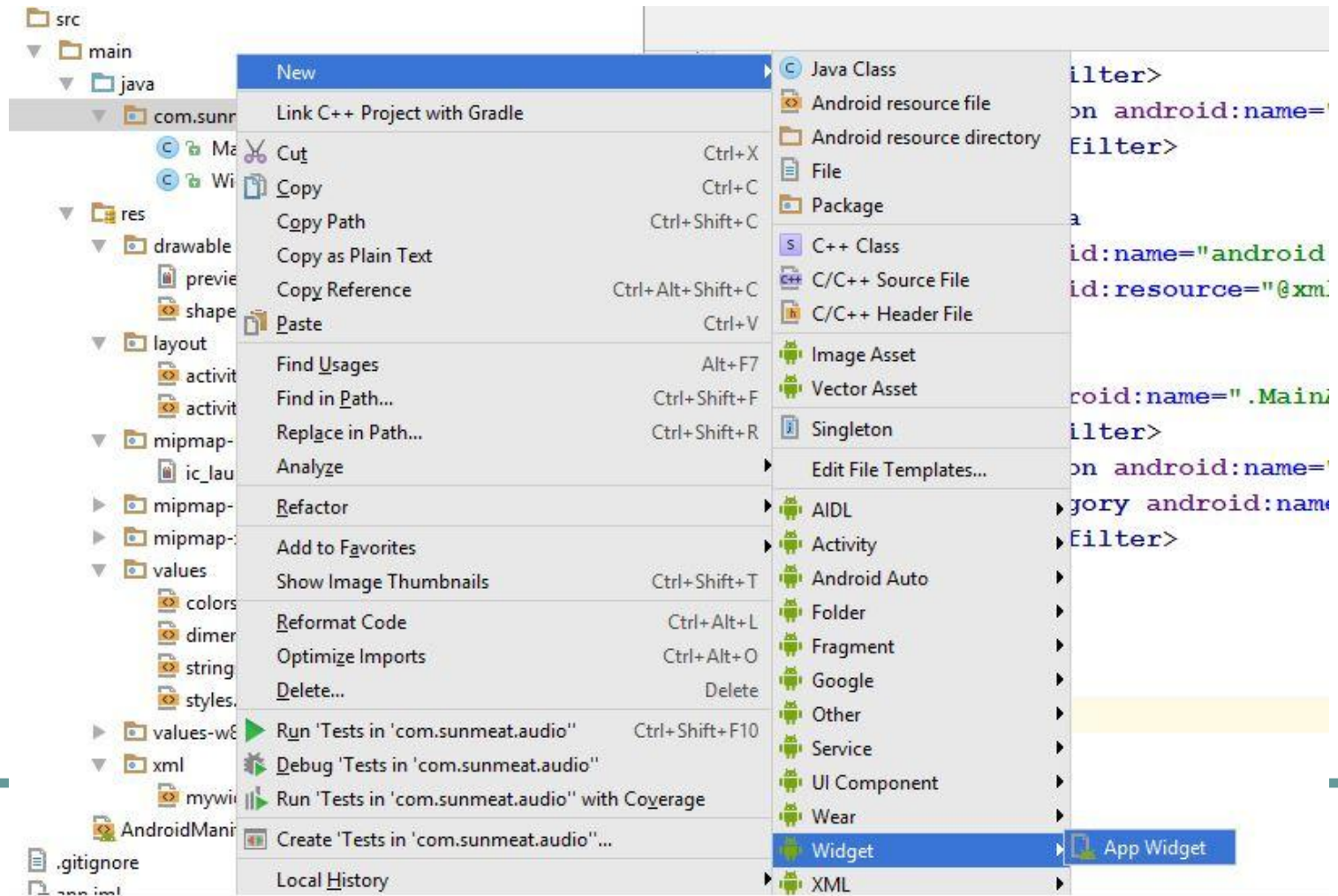
# Виджет + активити

- Добавляем с помощью шаблона новое пустое активити с названием **Widget1**
- В главном меню **Run > Edit Configurations... > Android App > app > Launch Options** возвращаем на место **Launch: Default Activity**
- Возможно, вновь придётся выполнить **Build > Clean Project**, затем **Build > Make Project**, опционально **Rebuild Project**, и наконец **Run > Run app...**

<https://git.io/v1SN5>


# Активити + два виджета

Вообще, добавить новый виджет в приложение не представляет особой сложности:




# Активити + два виджета

New Android Component

 **Configure Component**  
Android Studio

Creates a new App Widget



Class Name:

Placement:  ▼

Resizable (API 12+):  ▼

Minimum Width (cells):  ▼

Minimum Height (cells):  ▼

☐ Configuration Screen

# Практика

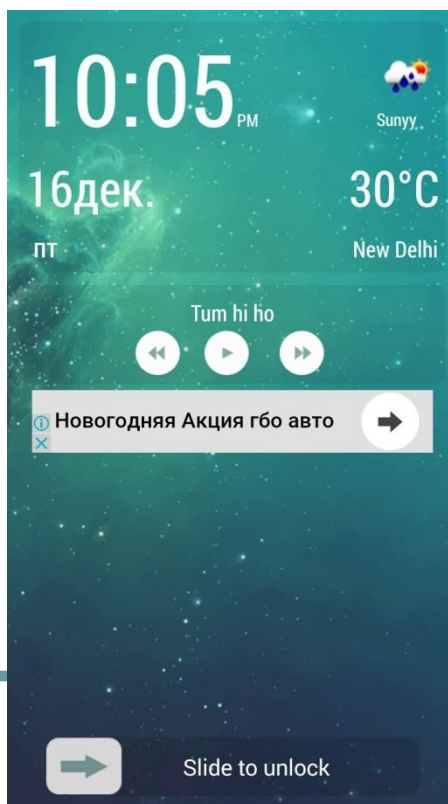
- Добавить Widget2 произвольного размера, home-screen only, без configuration screen
- Добавить Widget3 с configuration screen, протестировать установку виджетов
- Если версия ОС KitKat и ниже, создать виджет для keyguard (lock screen)

• The `widgetCategory` attribute declares whether your App Widget can be displayed on the home screen (`home_screen`), the lock screen (`keyguard`), or both. Only Android versions lower than 5.0 support lock-screen widgets. For Android 5.0 and higher, only `home_screen` is valid.



# Lollipop+ and LockScreen

Некоторые приложения, вроде Weather Lockscreen Widget, при установке на ОС Lollipop и выше, ведут себя так, будто действительно размещаются на экране блокировки. На самом деле, они лишь эмулируют поведение лок-скрина, реагируя на пробуждение программной разблокировкой.



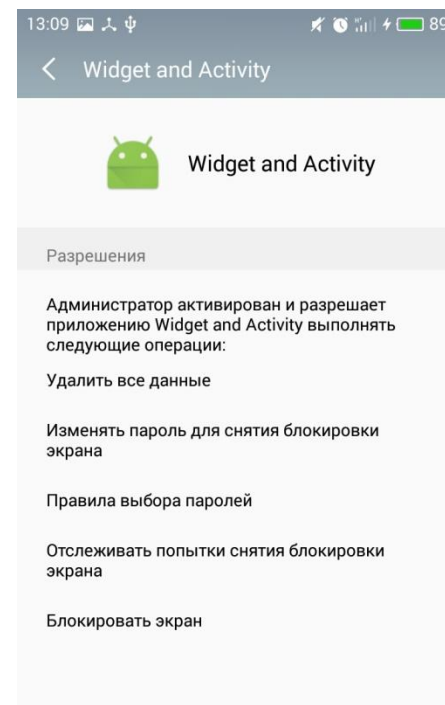
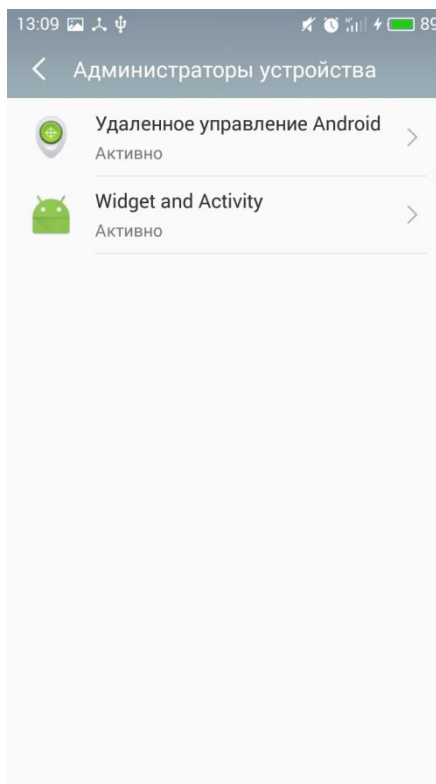
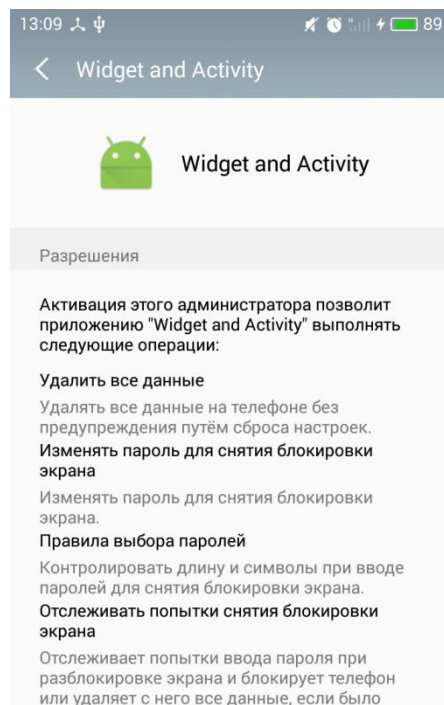
Receiver for SCREEN\_ON event:  
<http://stackoverflow.com/questions/9477922/android-broadcast-receiver-for-screen-on-and-screen-off>

Locking and unlocking an android device programmatically:  
<http://stackoverflow.com/questions/12723185/android-screen-lock-unlock-programmatically/12723269#12723269>



# Программная блокировка

<http://rdcworld-android.blogspot.com/2012/03/lock-phone-screen-programmatically.html>



<https://developer.android.com/guide/topics/admin/device-admin.html#lock>

# Программная разблокировка

<http://stackoverflow.com/questions/3793221/how-to-display-activity-when-the-screen-is-locked>

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    final Window wind = this.getWindow();
    Handler h = new Handler();
    h.postDelayed(new Runnable() {
        @Override
        public void run() {

            wind.addFlags(WindowManager.LayoutParams.FLAG_DISMISS_KEYGUARD);
            wind.addFlags(WindowManager.LayoutParams.FLAG_SHOW_WHEN_LOCKED);
            wind.addFlags(WindowManager.LayoutParams.FLAG_TURN_SCREEN_ON);

        }
    }, 5000);
}
```

# Класс MediaSession

При желании, можно разместить элементы управления mp3-плеера на экране блокировки, но для этого потребуется настройка объекта типа **MediaSession**:



- <http://sapandiwakar.in/android-media-player-lock-screen/>
- <https://developer.android.com/reference/android/media/session/MediaSession.html>
- <https://www.binpress.com/tutorial/using-android-media-style-notifications-with-media-session-controls/165>
- <https://android.googlesource.com/platform/developers/demos/+6768540171cd815002160bf76dcee1ae57b0fb28/MusicDemo/src/main/java/com/example/android/musicservicedemo/MediaNotification.java>

# Обновление при щелчке

<https://git.io/v17iB>

14:10   

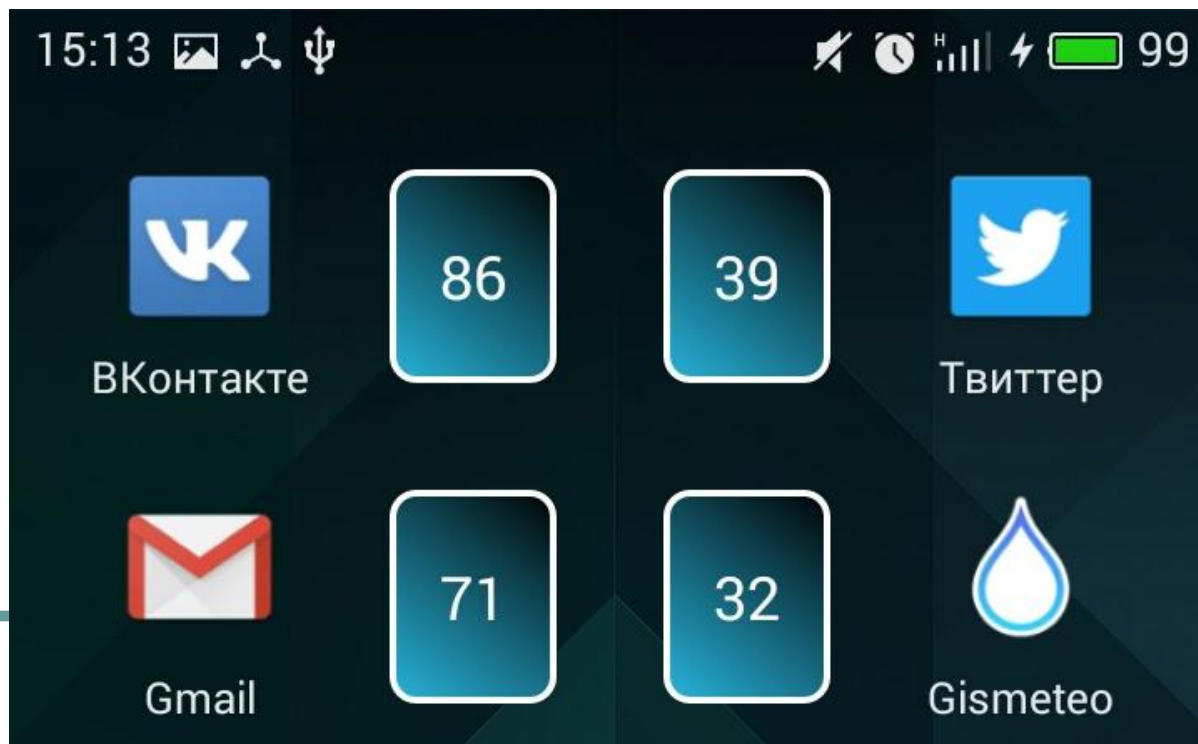
    96

55

# Обновление с помощью сервиса

- Создать файл UpdateWidgetService.java

<https://git.io/v17X7>



# Виджет «Курс валют»

<https://git.io/v1FTz>

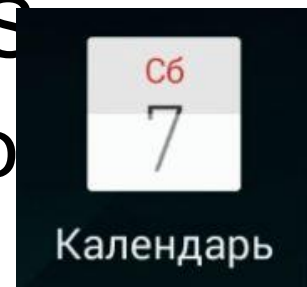


# Практика

- Добавить в виджет «Курс валют» отображение курса покупки доллара
- Создать виджет «Электронные часы», который будет отображать текущее время в формате ЧЧ:ММ:СС
- Создать виджет «Погода», который будет показывать прогноз погоды на сегодня (температуру, скорость ветра)

# Практика

- Создать виджет для просмотра картинок с сайта 9gag  
<http://9gag-rss.com/> Создать виджет для просмотра картинок с сайта 9gag  
<http://9gag-rss.com/>, цитат с башорга  
<http://bash.im/rss/> или любого другого сайта, предоставляющего RSS
- Создать виджет-календарь, который будет обновляться раз в день





# Дополнительные примеры

- <http://www.vogella.com/tutorials/AndroidWidgets/article.html>
- <https://www.sitepoint.com/how-to-code-a-n-android-widget/>
- <http://stacktips.com/tutorials/android/app-widgets-example-in-android>