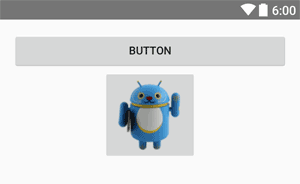
ImageButton (Кнопка-изображение)

[Общая информация](http://developer.alexanderklimov.ru/android/views/imagebutton.php#common)  
[Методы](http://developer.alexanderklimov.ru/android/views/imagebutton.php#methods)  
[Примеры](http://developer.alexanderklimov.ru/android/views/imagebutton.php#samples)  
[Щелчок](http://developer.alexanderklimov.ru/android/views/imagebutton.php#click)  
[Продолжительное нажатие](http://developer.alexanderklimov.ru/android/views/imagebutton.php#longclick)

Общая информация

Находится в разделе **Widgets**.

Компонент **ImageButton** представляет собой кнопку с изображением (вместо текста). По умолчанию **ImageButton** похож на обычную кнопку.



В режиме дизайна изображение на кнопке определяется атрибутом **android:src**

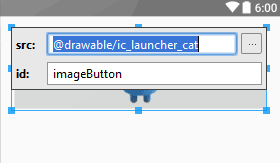
<ImageButton

android:id="@+id/imageButton"

...

android:src="@drawable/cat"/>

Можно сделать двойной щелчок, чтобы сразу установить нужное свойство.



Методы

Программно можно установить изображения через различные методы.

**setImageBitmap()**

Используется, чтобы указать в качестве изображения существующий экземпляр класса Bitmap

**setImageDrawable()**

Используется, чтобы указать в качестве изображения существующий экземпляр класса Drawable

**setImageResource()**

Используется, чтобы указать в качестве изображения существующий идентификатор ресурса (см. пример)

**setImageURI()**

Используется, чтобы указать в качестве изображения существующий адрес типа **Uri**. В некоторых случаях элемент кэширует изображение и после изменения изображения по прежнему выводит старую версию. Рекомендуется использовать инструкцию **setImageURI(null)** для сброса кэша и повторный вызов метода с нужным **Uri**

Примеры

С помощью метода **setImageURI()** можно обратиться к ресурсу типа **Drawable** по его идентификатору:

android.resource://[пакет]/[идентификатор ресурса]

Например, можно задать путь **Uri**:

Uri uri = Uri.parse("android.resource://ru.alexanderklimov.imagebuttondemo/" + R.drawable.cat);

Можно обратиться к ресурсу по его типу/имени:

android.resource://[пакет]/[тип ресурса]/[имя ресурса]

В этом случае код будет следующим:

Uri uri = Uri.parse("android.resource://ru.alexanderklimov.imagebuttondemo/drawable/cat");

imageButton.setImageUri(uri);

Щелчок

Как и у обычной кнопки, интерес представляет только щелчок. В нашем примере мы будем менять поочередно картинки на кнопке (нужно подготовить две картинки в папке **drawable**)

...

ImageButton imageButton;

boolean flag = true;

imageButton.setOnClickListener(new View.OnClickListener()

{

public void onClick(View v)

{

// меняем изображение на кнопке

if (flag)

imageButton.setImageResource(R.drawable.cutecat4);

else

// возвращаем первую картинку

imageButton.setImageResource(R.drawable.cutecat2);

flag = !flag;

}

});

Теперь при каждом щелчке изображение на кнопке будет циклически переключаться между двумя картинками.

Продолжительное нажатие

Кроме обычного щелчка, в Android есть особый вид нажатия на кнопку - продолжительное нажатие. Это событие происходит, когда пользователь нажимает и удерживает кнопку в течение одной секунды. Этот тип нажатия обрабатывается независимо от обычного щелчка.

Для обработки продолжительного нажатия нужно реализовать класс **View.OnLongClickListener** и передать его в метод **setOnLongClickListener()**. Класс **OnLongClickListener** имеет один обязательный метод **OnLongClick()**. В принципе это похоже на метод **OnClick()**, только имеет возвращаемое значение.

imageButton.setOnLongClickListener(new OnLongClickListener() {

@Override

public boolean onLongClick(View v) {

Toast toast = Toast.makeText(getApplicationContext(),

"Long Click",

Toast.LENGTH\_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();

return false;

}

});

Запустите проект и убедитесь, что при быстром нажатии ничего не происходит, а при более продолжительном нажатии всплывает сообщение.