ViewStub

Компонент **ViewStub** - это легкий **View**, не имеющий размеров. Находится в разделе **Expert**. Он ничего не рисует на экране, если его разместить на форме. Так зачем он нужен?

**ViewStub** выполняет роль заглушки, заменяя собой дочерние элементы внутри родительской разметки.

Возможно ваша разметка чересчур перегружена тяжёлыми элементами, которые редко используются. **ViewStub** позволят разгрузить разметку. Предположим, вы разработали экран с книжной полкой. Если у пользователя нет книг в коллекции, то нет особого смысла выводить TextView, Button и другие элементы с указанием авторов книги, названия и т.д. В нужный момент вы программно подгружаете невидимую разметку. По существу, это лёгкий элемент для отображения или сокрытия в иерархии **View**. Каждый **ViewStub** просто должен включать в себя атрибут **android:layout** для указания разметки, которую нужно отобразить.

Создадим новый проект и добавим **ViewStub** и другие компоненты.

<LinearLayout 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"

android:orientation="vertical"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Где ViewStub?" />

<Button

android:id="@+id/button"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:onClick="onClick"

android:text="Открыть ViewStub" />

<ViewStub

android:id="@+id/viewStub"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom"

android:inflatedId="@+id/inflate"

android:layout="@layout/viewstub\_layout" />

</LinearLayout>

Мы указали у **ViewStub** в атрибуте **android:layout** разметку **@layout/viewstub\_layout**. Создадим её в папке **res/layout**:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:id="@+id/textViewInViewStub"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Я TextView из ViewStub" />

<Button

android:id="@+id/buttonInViewStub"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:minWidth="100dp"

android:text="Я кнопка из ViewStub" />

</LinearLayout>

Теперь переходим к основной активности и напишем код для нажатия кнопки, который и будет выводить подготовленную разметку:

package ru.alexanderklimov.testapplication;

import ...

public class MainActivity extends ActionBarActivity {

private boolean mIsClicked = true;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

public void onClick(View v) {

if (mIsClicked) {

ViewStub viewStub = (ViewStub) findViewById(R.id.viewStub);

viewStub.inflate();

mIsClicked = false;

}

}

}

Булева переменная **mIsClicked** позволяет уберечься от повторного показа **ViewStub**. C помощью метода **inflate()** мы выводим разметку из файла **viewstub\_layout.xml** на экран вместо самого **ViewStub**. Обратите внимание на атрибут **android:inflatedId="@+id/inflate"** у **ViewStub** в первой разметке. Когда мы вызываем метод **inflate()** и **ViewStub** станет видимым, он больше не будет являться частью иерархии **View**, и ID для корня **View** заменится на указанный в атрибуте **android:inflatedId** идентификатор **inflate** (**android:id**, указанный для **ViewStub**, действует только до того момента, пока **ViewStub** не станет видимым). В этом легко убедиться. Добавим для кнопки из разметки **viewstub\_layout.xml** обработчик нажатия кнопки:

<Button

android:id="@+id/buttonInViewStub"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:minWidth="100dp"

android:onClick="onViewStubButtonClick"

android:text="Я кнопка из ViewStub" />

Напишем код для щелчка:

public void onViewStubButtonClick(View view) {

View testView = (View) findViewById(R.id.inflate);

testView.setBackgroundColor(Color.BLUE);

}

Когда мы нажимаем на первую кнопку, то выводим на экран контейнер **ViewStub**, который содержит ещё одну кнопку. Кроме того у нас появился новый идентификатор **inflate** и мы пробуем закрасить разметку в синий цвет. Если запустить проект, то увидим работающий пример. А если мы установим идентификатор самого **ViewStub** - **viewStub**, то получим ошибку.