[Получение всех ресурсов Drawable из объекта Resources](http://www.ohandroid.com/drawable-x43.html)

В моем проекте Android я хочу пропустить всю коллекцию ресурса Drawable . Обычно вы можете получить только определенный ресурс через свой идентификатор, используя что-то вроде:

InputStream is = Resources.getSystem().openRawResource(resourceId)

Тем не менее, я хочу получить все ресурсы Drawable где я **не буду** знать их идентификаторы заранее. Есть ли коллекция, которую я могу пропустить, или, возможно, способ получить список идентификаторов ресурсов, заданных ресурсами моего проекта?

Или, есть ли способ для меня в Java извлекать все значения свойств из статического класса R.drawable ?

* [Android-java- Как отсортировать список объектов по определенному значению внутри объекта](http://www.ohandroid.com/android-java-3.html)
* [Как инициализировать статический SparseArray](http://www.ohandroid.com/41118.html)

Solutions Collecting From Web of "Получение всех ресурсов Drawable из объекта Resources"

* [Как инициализировать статический SparseArray](http://www.ohandroid.com/41118.html)
* [Android-java- Как отсортировать список объектов по определенному значению внутри объекта](http://www.ohandroid.com/android-java-3.html)

Если вы обнаружите, что хотите сделать это, вы, вероятно, злоупотребляете системой ресурсов. Взгляните на активы и AssetManager если вы хотите перебирать файлы, включенные в ваш .apk.

Ладно, это немного похоже на хак-иш, но это то, что я придумал с помощью Reflection.(Обратите внимание, что resources – это экземпляр класса android.content.res.Resources .)

final R.drawable drawableResources = new R.drawable(); final Class<R.drawable> c = R.drawable.class; final Field[] fields = c.getDeclaredFields(); for (int i = 0, max = fields.length; i < max; i++) { final int resourceId; try { resourceId = fields[i].getInt(drawableResources); } catch (Exception e) { continue; } /\* make use of resourceId for accessing Drawables here \*/ }

Если у кого-то есть лучшее решение, которое лучше использует вызовы Android, о которых я, возможно, и не подозреваю, я бы очень хотел их увидеть!

Я использовал getResources (). GetIdentifier для сканирования последовательно названных изображений в папках ресурсов. Чтобы быть в безопасности, я решил кэшировать идентификаторы изображений, когда активность создается впервые:

private void getImagesIdentifiers() { int resID=0; int imgnum=1; images = new ArrayList<Integer>(); do { resID=getResources().getIdentifier("img\_"+imgnum, "drawable", "InsertappPackageNameHere"); if (resID!=0) images.add(resID); imgnum++; } while (resID!=0); imageMaxNumber=images.size(); }

Добавьте изображение с именем aaaa и другим именем zzzz, затем выполните следующие действия:

public static void loadDrawables() { for(long identifier = (R.drawable.aaaa + 1); identifier <= (R.drawable.zzzz - 1); identifier++) { String name = getResources().getResourceEntryName(identifier); //name is the file name without the extension, indentifier is the resource ID } }

Это сработало для меня.

Я думаю, что код отражения будет работать, но я не понимаю, зачем вам это нужно.

Ресурсы в Android статичны, если приложение установлено, поэтому вы можете иметь список ресурсов или массив. Что-то вроде:

<string-array name="drawables\_list"> <item>drawable1</item> <item>drawable2</item> <item>drawable3</item> </string-array>

И из вашей Activity вы можете получить ее, выполнив:

getResources().getStringArray(R.array.drawables\_list);

Вы должны использовать папку Raw и AssetManager, но если вы хотите использовать чертежи, потому что почему бы и нет, вот как …

Предположим, у нас есть очень длинный список файлов JPG-чертежей, и мы хотим получить все идентификаторы ресурсов без боли при поиске по одному (R.drawable.pic1, R.drawable.pic2, … и т. Д.).

//first we create an array list to hold all the resources ids ArrayList<Integer> imageListId = new ArrayList<Integer>(); //we iterate through all the items in the drawable folder Field[] drawables = R.drawable.class.getFields(); for (Field f : drawables) { //if the drawable name contains "pic" in the filename... if (f.getName().contains("image")) imageListId.add(getResources().getIdentifier(f.getName(), "drawable", getPackageName())); } //now the ArrayList "imageListId" holds all ours image resource ids for (int imgResourceId : imageListId) { //do whatever you want here }

Я принял Мэтта Хаггинса отличный ответ и реорганизовал его, чтобы сделать его более общим:

public static void loadDrawables(Class<?> clz){ final Field[] fields = clz.getDeclaredFields(); for (Field field : fields) { final int drawableId; try { drawableId = field.getInt(clz); } catch (Exception e) { continue; } /\* make use of drawableId for accessing Drawables here \*/ } }

Применение:

loadDrawables(R.drawable.class);

Просто сделайте это:

Field[] declaredFields = (R.drawable.class).getDeclaredFields();

OP нуждался в чертежах, и мне нужны макеты. Это то, что я придумал для макетов.name.startsWith business позволяет мне игнорировать сгенерированные системой макеты, поэтому вам может понадобиться немного name.startsWith . Это должно работать для любого типа ресурса, изменяя значение clz .

public static Map<String,Integer> loadLayouts(){ final Class<?> clz = R.layout.class; Map<String,Integer> layouts = new HashMap<>(); final Field[] fields = clz.getDeclaredFields(); for (Field field : fields) { String name = field.getName(); if ( !name.startsWith("abc\_") && !name.startsWith("design\_") && !name.startsWith("notification\_") && !name.startsWith("select\_dialog\_") && !name.startsWith("support\_") ) { try { layouts.put(field.getName(), field.getInt(clz)); } catch (Exception e) { continue; } } } return layouts; }

**ИСПОЛЬЗУЙТЕ ЭТО МОЙ КОД**

R.drawable drawableResources = new R.drawable(); Class<R.drawable> c = R.drawable.clas