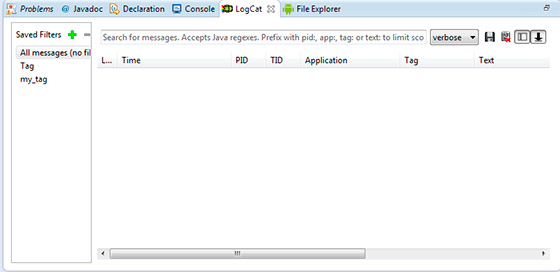
Логгирование в андроид-приложениях

[admin](http://androiddocs.ru/author/admin/) | 23.06.2014

[0 Comment](http://androiddocs.ru/loggirovanie-v-android-prilozheniyax/#comments)

При разработке андроид-приложений очень часто нужно видеть и понимать, какое значение имеет то или иное свойство объекта, определенная переменная и т.д. Чтобы получать такую информацию в разработке используется логгирование. Логгирование использует простые методы, которые выводят нужные данные в консоли Eclipse (ADT). Выглядит это так:



Здесь нас интересует, прежде всего, вкладка **LogCat**. Если она отсутствует, то включите ее — выберите в меню **Window -> Show View -> Other -> Android-> Logcat**.

Теперь в нужном месте кода достаточно добавить одну строку:

|  |  |
| --- | --- |
| 1 | Log.d("my\_tag", "price = " + val); |

которая в процессе работы выведет в консоль значение переменной **price**. Лог будет помечен тэгом **«my\_tag»**, чтобы его было легче найти в куче других логов (вы можете выбрать произвольное имя). Кстати, более правильно имя тега задавать как константу (свойство класса) и подставлять в параметры метода только ее имя:

|  |  |
| --- | --- |
| 1 | private static final String TAG = "my\_tag"; |

и затем

|  |  |
| --- | --- |
| 1 | Log.d(TAG, "price = " + val); |

В данном случае мы используем статический метод **d(String tag, String msg)** класса **Log**. Этот метод задает уровень «DEBUG» для лога. Есть несколько таких уровней: **ERROR, WARN, INFO, DEBUG, VERBOSE** (перечислены от меньшего к большему). Если вы в консоли будете просматривать ошибки только уровня ERROR, то фильтр не покажет вам сообщения более верхних уровней, например, **INFO** и**DEBUG**. Фильтр по уровню **DEBUG** покажет также ERROR, WARN и INFO.

Давайте создадим простой пример с логом. При запуске приложения (запуске главного активити) будем писать сообщение в консоль.

Файл layout-a — **res/layout/activity\_main.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"  android:gravity="center"  android:orientation="vertical" >    <TextView  android:id="@+id/textView1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/hello\_world" />    </LinearLayout> |

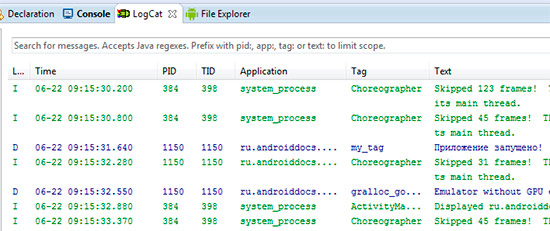
Строковые ресурсы — **res/values/strings.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <?xml version="1.0" encoding="utf-8"?>  <resources>    <string name="app\_name">Logs</string>  <string name="hello\_world">Hello world!</string>  <string name="action\_settings">Settings</string>    </resources> |

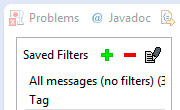
**MainActivity.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | package ru.androiddocs.logs;    import android.app.Activity;  import android.os.Bundle;  import android.util.Log;    public class MainActivity extends Activity {  // константа для имени лога  private static final String TAG = "my\_tag";    @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  // пишем сообщение в логи  Log.d(TAG, "Приложение запущено!");  }  } |

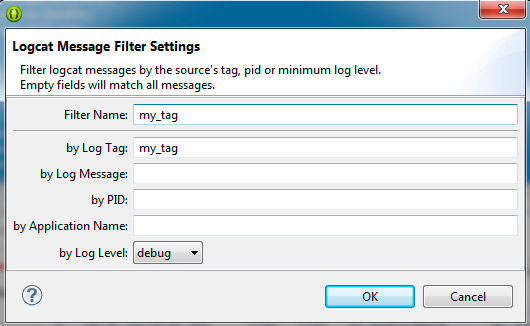
Теперь, если мы запустим приложение, то увидим в куче логов и наше сообщение (первое сообщение с синим цветом шрифта).



Для удобства можно настроить фильтрацию по нашему лог-тегу. Нажмите на плюсик для добавления нового фильтра.



В открывшемся окне добавляем название фильтра, тег для фильтрации и уровень «debug».



Теперь все наши логи с тегом «my\_tag» будут отображаться в отдельной секции.

