**Как создать свое меню для Android приложения?**

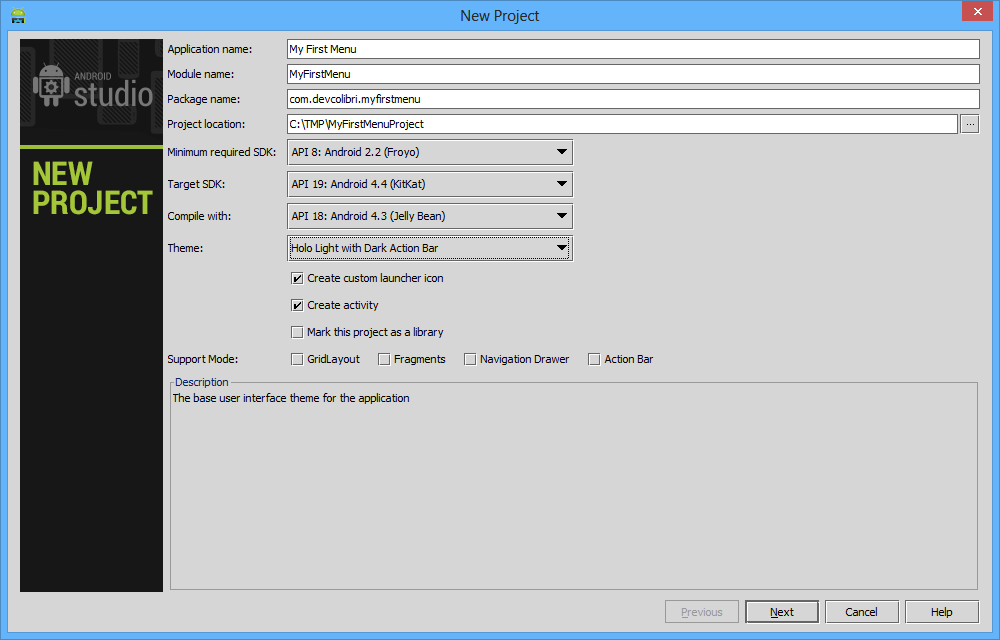
Шаг 0: Android Studio в действии

Решил я начать использовать **Android Studio 0.3.2**все же прошло уже не мало времени с момента релиза. Так вот этот урок будет в Android Studio 0.3.2 вот ссылка на студию: <http://developer.android.com/sdk/installing/studio.html>



**Шаг 1: Создаем проект**

Запускаем студию и нажимаем **Create New Project**:

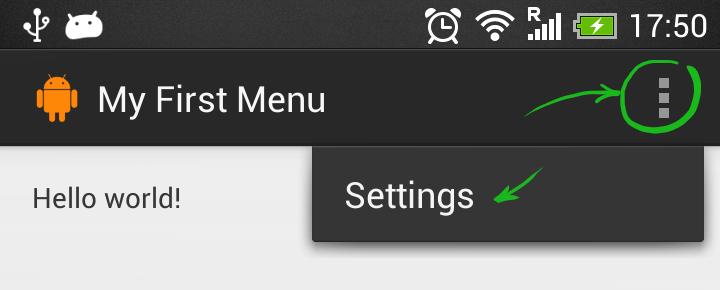


Как видите мы настроили, что **Minimum required SDK = Android 2.2** — это значит, что приложение будет работать начиная с Android 2.2 и на самой новой на данный момент Android 4.4

**Шаг 2: Разбор стандартного проекта**

Теперь пора перейти к решению поставленой задачи, а именно создать свое меню для приложения.

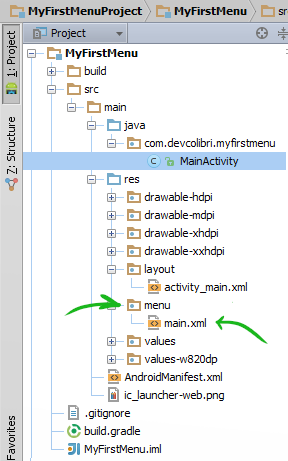
Если запустить только что созданный проект, то вы увидите, что он имеет такое меню:



Именно вместо Settings мы научимся добавлять свои элементы.

Давате разберемся как же это меню там появилось.

Вот структура проекта:

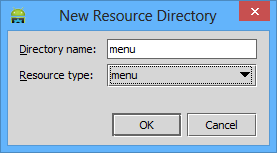


Как вы видите в папке **res**есть папка **menu** — она предназначена для хранинение XML-дескрипторов меню.

**Шаг 3: Создаем меню**

Теперь давайте создадим свое меню.

Для начало создайте папку menu в папке res если её у вас нет.



А теперь создадим в этой папке файл **main.xml**со следующим содержимым:

|  |
| --- |
| <menu xmlns:android="<http://schemas.android.com/apk/res/android>"      xmlns:app="<http://schemas.android.com/apk/res-auto>"      xmlns:tools="<http://schemas.android.com/tools>"      tools:context=".MainActivity" >        <item android:id="@+id/action\_settings"          android:title="@string/action\_settings"          android:orderInCategory="100"          app:showAsAction="never" />  </menu> |

Как можно увидеть в примере высше, то там есть элемент **item**идавай разберем его атрибуты:

android:**title** — это тот текст, который будет отображаться пользователю. В нашем случае — это **Settings**.

android:**orderInCategory** — указывает позицию в списке меню, так как там указанно 100, то скорее всего он всегда будет последним.

app:**showAsAction**— говорит как будет отображаться элемент. Может принимать следующие значения:

always — всегда будет виден, если места не хватает, заголовок будет показан не полностью;  
ifRoom — будет виден, только если для него есть место;  
never — элемент никогда не будет виден, для того чтобы его увидеть нужно раскрыть Меню.;  
withText -будет показываться только с его заголовком;  
collapseActionView — может сворачиваться в кнопку или разворачиваться на всю ширину ActionBar по нажатию;

В нашем примере используется **never**поэтому мы видим **Settings**, когда раскрываем меню.

**Шаг 4: Добавляем меню в приложение**

Для этого в главной активности, в моем случае это **MainActivity**.java нужно унаследоваться от **ActionBarActivity**и  переопределить метод **onCreateOptionsMenu**(Menu menu):

|  |
| --- |
| package com.devcolibri.myfirstmenu;    import android.os.Bundle;  import android.support.v7.app.ActionBarActivity;  import android.view.Menu;    public class MainActivity extends ActionBarActivity {        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);      }        @Override      public boolean onCreateOptionsMenu(Menu menu) {          getMenuInflater().inflate(R.menu.main, menu);          return true;      }    } |

В строке 16 мы переопределили этот метод и на 17-й строке мы указываем наше меню. Обратите внимение, что R.menu.**main —**это имя файла что мы создали в шаге 3 main.xml.

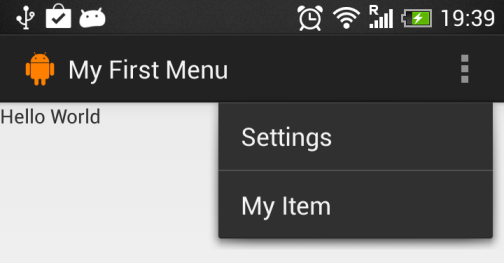
После этого мы увидим наше меню.

**Шаг 5: Добавляем новый элемент и действие на него**

новый элемент в main.xml меню:

|  |
| --- |
| <menu xmlns:android="<http://schemas.android.com/apk/res/android>"      xmlns:app="<http://schemas.android.com/apk/res-auto>"      xmlns:tools="<http://schemas.android.com/tools>"      tools:context=".MainActivity" >        <item android:id="@+id/action\_settings"          android:title="@string/action\_settings"          android:orderInCategory="100"          app:showAsAction="never" />        <item android:id="@+id/action\_my\_item"          android:title="@string/action\_my\_item"          android:orderInCategory="100"          app:showAsAction="never" />  </menu> |

Теперь меню выглядит так:



Теперь давайте добавим обработчик на эти пункты меню, для это зайходим в наше MainActivity и переопределяем метод **onOptionsItemSelected**(MenuItem item):

|  |
| --- |
| @Override  public boolean onOptionsItemSelected(MenuItem item) {      switch (item.getItemId()) {          case R.id.action\_settings:              Toast.makeText(getApplicationContext(),                      "You selected Settings", Toast.LENGTH\_LONG).show();              return true;            case R.id.action\_my\_item:              Toast.makeText(getApplicationContext(),                      "You selected My Item", Toast.LENGTH\_LONG).show();              return true;      }      return super.onOptionsItemSelected(item);  } |

После этого на нажатие по пунктам меню мы будем выводит сообщение.

**Шаг 6: Запуск и проверка меню**

Давайте запустим и проверим наше меню:

