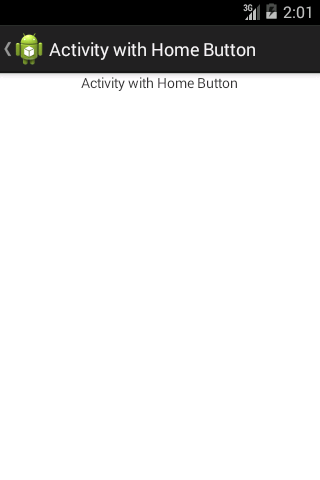
Action Bar Добавляем кнопку Home

Кнопка **Home** в Action Bar-е позволяет вернуться на домашний/главный/предыдущий экран приложения или любой другой. На самом деле, вы вольны обрабатывать клик так, как вам захочется, и можете добавить целевое активити такое, какое будет уместно в данном контексте. По этой причине название Home несколько узко, однако оно подразумевает, что переход на главный/родительский/предыдущий экран — это все же удобнее и понятнее для навигации.

Кнопка Home обычно расположена слева в Action Bar-е и включает в себя угловую скобку **<** и название текущего Активити.



Создадим простое приложение с двумя экранами/Активити. В первом будет добавлена кнопка, по которой мы можем кликнуть и перейти на второй экран. Приложение будет создано для API >= 9 (будем использовать **Support Library**).

Добавим строковые ресурсы **res/values/strings.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <resources>        <string name="app\_name">HomeButton</string>      <string name="activity\_with\_home">Activity with Home Button</string>      <string name="main\_activity">Main Activity</string>      <string name="change\_activity">Change Activity</string>      <string name="home\_activity">Home Activity</string>    </resources> |

[LinearLayout](http://androiddocs.ru/linearlayout/) для первого Активити — **res/layout/activity\_main.xml**

|  |
| --- |
| <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:gravity="top|center"      tools:context=".MainActivity">        <TextView          android:text="@string/main\_activity"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content" />        <Button          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:text="@string/change\_activity"          android:id="@+id/button"          android:onClick="changeActivity"          />    </LinearLayout> |

Не забудьте добавить для кнопки атрибут **android:onClick**, который будет отсылать обработку кликов на метод **changeActivity()**.

Код **MainActivity.java**

|  |
| --- |
| package ru.androiddocs.homebutton;    import android.content.Intent;  import android.support.v7.app.ActionBarActivity;  import android.os.Bundle;  import android.view.View;      public class MainActivity extends ActionBarActivity {        Intent mIntent;        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);      }        public void changeActivity(View v) {          mIntent = new Intent(this, ActivityWithHome.class);          startActivity(mIntent);      }  } |

Код простой. В **onCreate()** стандартные вещи. В методе **changeActivity()** мы создаем новый интент со вторым Активити и переходим на **ActivityWithHome** с помощью метода **startActivity()**, принимающим в качестве параметра созданный Intent.

Создадим layout для второго Активити — **res/layout/activity\_with\_home.xml**

|  |
| --- |
| <?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:gravity="top|center">        <TextView          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:text="@string/activity\_with\_home"          android:id="@+id/textView" />  </LinearLayout> |

Код второго Активити — **ActivityWithHome.java**

|  |
| --- |
| package ru.androiddocs.homebutton;    import android.content.Intent;  import android.os.Bundle;  import android.support.v7.app.ActionBar;  import android.support.v7.app.ActionBarActivity;  import android.view.MenuItem;    public class ActivityWithHome extends ActionBarActivity {        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_with\_home);            ActionBar actionBar = getSupportActionBar();          actionBar.setHomeButtonEnabled(true);          actionBar.setDisplayHomeAsUpEnabled(true);      }        @Override      public boolean onOptionsItemSelected(MenuItem item) {          switch (item.getItemId()) {              case android.R.id.home:                  startActivity(new Intent(this, MainActivity.class));                  return true;              default:                  return super.onOptionsItemSelected(item);          }      }  } |

Как видите, мы добавляем кнопку **Home** в методе **onCreate()** программно. Сначала получаем доступ до **ActionBar**:

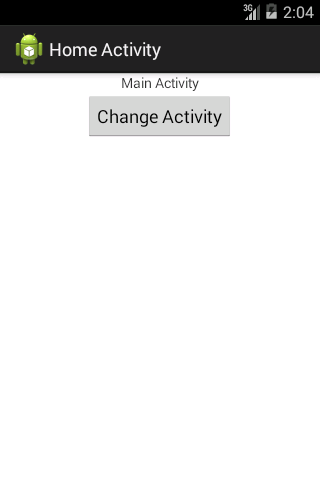
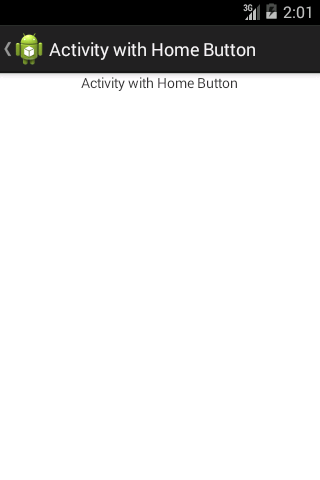
|  |
| --- |
| ActionBar actionBar = getSupportActionBar(); |

Для новых API можно использовать метод **getActionBar()**. Затем включаем кнопку —**setHomeButtonEnabled()**, передавая в качестве параметра **true**. Метод **setDisplayHomeAsUpEnabled()** включает/выключает отображение навигации и добавляет угловую скобку.

В методе **onOptionsItemSelected()** мы обрабатываем нажатие на кнопку **Home** — придет значение **android.R.id.home**. В этом блоке мы запускаем вновь наше главное Активити.

Не забудьте добавить в файл манифеста информацию о новом активити:

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"      package="ru.androiddocs.homebutton" >        <application          android:allowBackup="true"          android:icon="@drawable/ic\_launcher"          android:label="@string/app\_name"          android:theme="@style/AppTheme" >          <activity              android:name=".MainActivity"              android:label="@string/home\_activity" >              <intent-filter>                  <action android:name="android.intent.action.MAIN" />                    <category android:name="android.intent.category.LAUNCHER" />              </intent-filter>          </activity>          <activity              android:name=".ActivityWithHome"              android:label="@string/activity\_with\_home">          </activity>      </application>  </manifest> |

Также можно настроить «автоматическое» определение на какое «родительское» Активити мы должны перейти по нажатию кнопки Home. Например, если вы просматриваете список [ListView](http://androiddocs.ru/listview-sozdaem-prostejshij-spisok-s-obrabotkoj-vybora-punkta/" \t "_blank), а затем переходите на просмотр отдельного пункта, то весь список в данном случае и будет родителем. Для этого в блоке **case** нужно поместить такой код (пример для нашего приложения):

|  |  |
| --- | --- |
| 1  2  3 | case android.R.id.home:      NavUtils.navigateUpFromSameTask(this);      return true; |

А в файле манифеста для второго Активити прописать атрибут с родительским Активити:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <activity      android:name=".ActivityWithHome"      android:parentActivityName=".MainActivity" >      android:label="@string/activity\_with\_home">      <meta-data          android:name="android.support.PARENT\_ACTIVITY"          android:value=".MainActivity" />  </activity> |

Атрибут здесь содержит описание для версий API <= 4.

Если вы запустите сейчас обновленный код, то получите тот же результат, что и при «ручном» создании Intent-объекта.