

STARTING ANOTHER ACTIVITY

Sources:

<http://developer.android.com/training/basics/firstapp/starting-activity.html#ReceiveIntent>

An [Intent](#) is an object that provides runtime binding between separate components (such as two activities). The [Intent](#) represents an app's "intent to do something." You can use intents for a wide variety of tasks, but most often they're used to start another activity. For more information, see [Intents and Intent Filters](#).

An intent is an abstract description of an operation to be performed. It can be used with [startActivity](#) to launch an [Activity](#), [broadcastIntent](#) to send it to any interested [BroadcastReceiver](#) components, and [startService\(Intent\)](#) or [bindService\(Intent, ServiceConnection, int\)](#) to communicate with a background [Service](#).

Example 1

Layout

```
<RelativeLayout
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:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"
        android:gravity="center"
        tools:context=".MainActivity">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/bText"
        android:onClick="NewActivity"/>
</RelativeLayout>
```

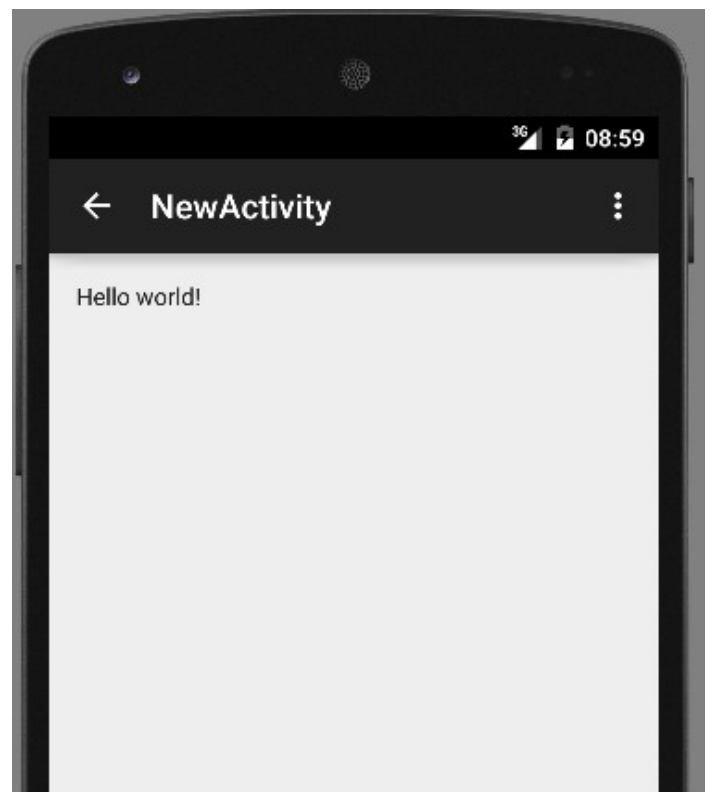
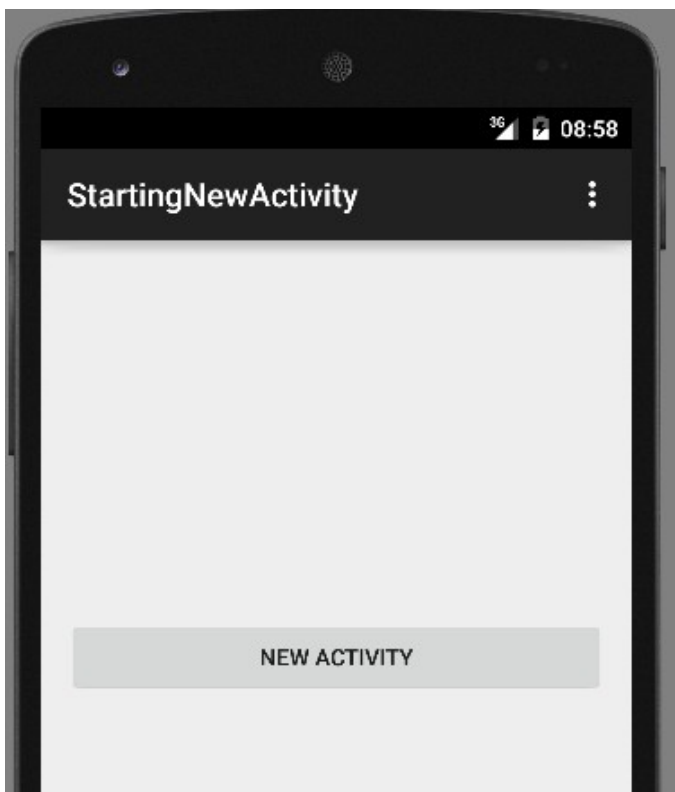
Strings

```
<resources>
    <string name="app_name">StartingNewActivity</string>
    <string name="action_settings">Settings</string>
    <string name="bText">New Activity</string>
</resources>
```

Create a new activity. Select File → New → Activity → Blank Activity. Give it a name. For example, NewActivity. Select its hierarchical parent (“...” button).

Modify old activity class (default MainActivity):

```
public void NewActivity(View view) {  
    Intent intent = new Intent(this, NewActivity.class);  
    startActivity(intent);  
}
```



Example 1

Layout (activity_new.xml)

```
<RelativeLayout  
    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:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context="com.example.startingnewactivity.app.NewActivity">
    <EditText
        android:id="@+id/et"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />
</RelativeLayout>

```

An [Intent](#) can carry data types as key-value pairs called *extras*. The [putExtra\(\)](#) method takes the key name in the first parameter and the value in the second parameter.

For the next activity to query the extra data, you should define the key for your intent's extra using a public constant. It's generally a good practice to define keys for intent extras using your app's package name as a prefix. This ensures the keys are unique, in case your app interacts with other apps.

Add a class member to MainActivity class:

```

int x = 5;
public final static String EXTRA_MESSAGE =
    "com.example.startingnewactivity.MESSAGE";

```

Modify NewActivity() method:

```

public void NewActivity(View view) {
    Intent intent = new Intent(this, NewActivity.class);
    String message = String.valueOf(x).toString();
    intent.putExtra(EXTRA_MESSAGE, message);
    startActivity(intent);
}

```

Add a class member to NewActivity class:

```

EditText et;

```

Modify its onCreate() method:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_new);

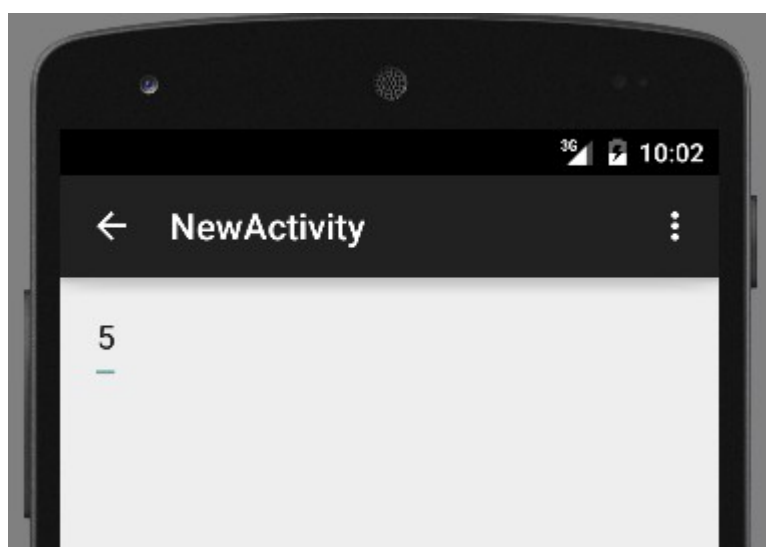
    et = (EditText) findViewById(R.id.et);
}
```

Receive the intent

Every [Activity](#) is invoked by an [Intent](#), regardless of how the user navigated there. You can get the [Intent](#) that started your activity by calling [getIntent\(\)](#) and retrieve the data contained within the intent.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_new);
    Intent intent = getIntent();
    String message =
intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
    et = (EditText) findViewById(R.id.et);
    et.setText(message);
}
```

Compile the application.



You can put not only String:

<u>Intent</u>	<u>putExtra(String name, double[] value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, int value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, CharSequence value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, char value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, Bundle value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, Parcelable[] value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, Serializable value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, int[] value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, float value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, byte[] value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, long[] value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, Parcelable value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, float[] value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, long value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, String[] value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, boolean value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, boolean[] value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, short value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, double value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, short[] value)</u> Add extended data to the intent.
<u>Intent</u>	<u>putExtra(String name, String value)</u> Add extended data to the intent.

Intent	putExtra(String name, byte value) Add extended data to the intent.
Intent	putExtra(String name, char[] value) Add extended data to the intent.
Intent	putExtra(String name, CharSequence[] value) Add extended data to the intent.

Receiving an answer

activity_main.xml (layout)

```
<RelativeLayout
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:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
        android:gravity="center"
        tools:context=".MainActivity">
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/bText"
        android:onClick="NewActivity"/>
    <EditText
        android:layout_below="@+id/button"
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</RelativeLayout>
```

Modify MainActivity class. Add a class member

```
EditText et;
int requestCode = 1;
```

Modife onCreate() method (MainActivity):

```
et = (EditText) findViewById(R.id.editText);
```

Add a button to new layout:

```
<RelativeLayout
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:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"

    tools:context="com.example.startingnewactivity.app.NewActivity">
    <EditText
        android:id="@+id/et"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/buttonNew"
        android:layout_below="@+id/et"
        android:onClick="onMain"/>

</RelativeLayout>
```

Add onMain() method to NewActivity class:

```
public void onMain(View view) {
    String text = et.getText().toString();
    Intent intent = new Intent();
    intent.putExtra("data", text);
    setResult(RESULT_OK, intent);
    finish();
}
```

Add onActivityResult() method to ActivityMain class:

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (data == null) {return;}
    if(requestCode == 1)
        if(resultCode == RESULT_OK){
            String text = data.getStringExtra("data");
            et.setText(text);
        }
}
```

