

# How to start new activity on button click

In an Android application, how do you start a new activity (GUI) when a button in another activity is clicked, and how do you pass data between these two activities?

 android   android-intent   android-activity   android-button   android-lifecycle

edited Dec 22 '15 at 17:07



MiguelHincapieC  
2,814 2 21 51

asked Nov 15 '10 at 15:43



Adham  
20.7k 79 191 307

## 17 Answers

Easy.

```
Intent myIntent = new Intent(CurrentActivity.this, NextActivity.class);
myIntent.putExtra("key", value); //Optional parameters
CurrentActivity.this.startActivity(myIntent);
```

Extras are retrieved on the other side via:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    Intent intent = getIntent();
    String value = intent.getStringExtra("key"); //if it's a string you stored.
}
```

Don't forget to add your new activity in the AndroidManifest.xml:

```
<activity android:label="@string/app_name" android:name="NextActivity"/>
```

edited May 18 '16 at 17:34



DenisKolodin  
2,927 25 33

answered Nov 5 '10 at 15:51



Emmanuel  
12.9k 4 35 62

- |    |   |
|----|---|
| 11 | Where is the button click part? (button click → transition to next activity) – Jonny Nov 5 '12 at 13:25   |
| 2  | @Jonny: Here's an example of a button click. <a href="http://stackoverflow.com/a/7722428/442512">stackoverflow.com/a/7722428/442512</a> – Emmanuel Nov 5 '12 at 19:49 |
| 5  | Is there any difference between <code>CurrentActivity.this.startActivity(myIntent)</code> and <code>startActivity(myIntent)</code> ? – Confuse Sep 19 '14 at 10:31    |
| 1  | Yeah, easy lol. There is more code missing than code actually typed. Where is all xml interface and .java code missing? Downvote – Liquid Core Jun 21 '15 at 12:50    |
| 50 | Liquid, would you like him to pack it into an apk, too? ;) – Casey Murray May 11 '16 at 5:26  |

Create an intent to a ViewPerson activity and pass the PersonID (for a database lookup, for example).

```
Intent i = new Intent(getBaseContext(), ViewPerson.class);
i.putExtra("PersonID", personID);
startActivity(i);
```

Then in ViewPerson Activity, you can get the bundle of extra data, make sure it isn't null (in case if you sometimes don't pass data), then get the data.

```
Bundle extras = getIntent().getExtras();
if(extras != null)
{
    personID = extras.getString("PersonID");
}
```

Now if you need to share data between two Activities, you can also have a Global Singleton.

```
public class YourApplication extends Application
{
    public SomeDataClass data = new SomeDataClass();
}
```

Then call it in any activity by: