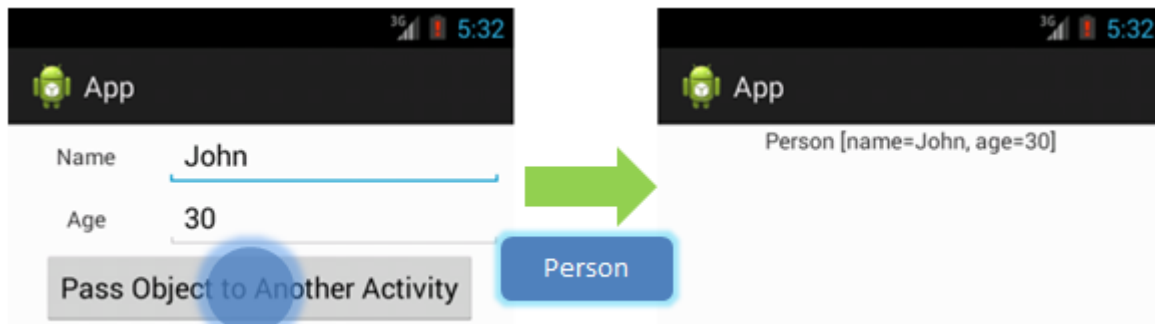


## Android – Passing Java Object to Another Activity



We can pass simple “primitive” data type (String, int, double,...ect) to activities [Passing Data to Another Activities](#). However, passing an object is a little bit different. Here we will see how to pass “Java” objects from current activity to another activities.

### Objectives:

- How to pass Java Objects to other activities?
- How to put the data in Intent object?

### Environment & Tools:

- Android Developer Tools (ADT) (or Eclipse + ADT plugin)
- AVD Nexus S Android 4.3 “emulator”
- Min SDK 8

### ( 1 ) Create Android Application

- **File >> New >> Android Application**
- Enter Project name: **android-pass-object-to-activity**
- Package: **com.hmkcode.android**
- Keep other default selections, go **Next** till you reach **Finish**

Before start creating the App we need to create POJO class “Person” which we will use to send object from one activity to another. Notice that the class is implementing Serializable interface.

```

2
3 import java.io.Serializable;
4
5 public class Person implements Serializable{
6
7     private static final long serialVersionUID = 1L;
8
9     private String name;
10    private int age;
11
12    // getters & setters....
13
14    @Override
15    public String toString() {
16        return "Person [name=" + name + ", age=" + age + "]";
17    }
18 }

```

## ( 2 ) Two Layouts for Two Activities

- Define the layout for the first “main” activity “object sender and the second “another” activity “object receiver”.

### ■ res/layout/activity\_main.xml

- The layout contains two edit text to enter person name and age.Also, a button to pass person object to another activity.

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical"
6   tools:context=".MainActivity" >
7
8   <LinearLayout
9       android:layout_width="fill_parent"
10      android:layout_height="wrap_content"
11      android:orientation="horizontal">
12      <TextView
13          android:id="@+id/tvName"
14          android:layout_width="100dp"
15          android:layout_height="wrap_content"
16          android:layout_gravity="center"
17          android:gravity="center_horizontal"
18          android:text="Name" />
19
20      <EditText
21          android:id="@+id/etName"
22          android:layout_width="wrap_content"
23          android:layout_height="wrap_content"
24
25          android:ems="10" >
26          <requestFocus />
27      </EditText>
28  </LinearLayout>
29
30  <LinearLayout
31      android:layout_width="fill_parent"
32      android:layout_height="wrap_content"
33      android:orientation="horizontal">
34      <TextView
35          android:id="@+id/tvAge"
36          android:layout_width="100dp"
37          android:layout_height="wrap_content"
38          android:layout_gravity="center"
39          android:gravity="center_horizontal"

```

```

40         android:text="Age" />
41     <EditText
42         android:id="@+id/etAge"
43         android:layout_width="wrap_content"
44         android:layout_height="wrap_content"
45         android:ems="10" />
46 </LinearLayout>
47
48 <Button
49     android:id="@+id/btnPassObject"
50     android:layout_width="wrap_content"
51     android:layout_height="wrap_content"
52     android:layout_gravity="center_horizontal"
53     android:text="Pass Object to Another Activity" />
54
55 </LinearLayout>

```

#### ■ res/layout/activity\_another.xml

- The layout contains one TextView to display the object sent from the *sender* activity.

```

1  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical"
6      >
7
8      <TextView
9          android:id="@+id/tvPerson"
10         android:layout_height="wrap_content"
11         android:layout_width="fill_parent"
12         android:layout_gravity="center"
13         android:gravity="center_horizontal"
14     />
15
16 </LinearLayout>

```

## ( 3 ) Two Activity Classes

#### ■ src/com/hmkcode/android/ActivityMain.java

- **MainActivity.java** is a simple "*sender*" activity class implementing **OnClickListener** to handle button click event.
- When user click on the button we call **onClick**
- Within the **onClick** method we create the **intent**, **Person object**, put person into the intent and pass it to **startActivity(intent)** method.

```

1  package com.hmkcode.android;
2
3  import android.os.Bundle;
4  import android.app.Activity;
5  import android.content.Intent;
6  import android.view.View;
7  import android.view.View.OnClickListener;
8  import android.widget.Button;
9  import android.widget.EditText;
10
11 public class MainActivity extends Activity implements OnClickListener {
12
13     Button btnPassObject;
14     EditText etName, etAge;
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {

```

```

17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19
20         btnPassObject = (Button) findViewById(R.id.btnPassObject);
21         etName = (EditText) findViewById(R.id.etName);
22         etAge = (EditText) findViewById(R.id.etAge);
23
24         btnPassObject.setOnClickListener(this);
25     }
26
27     @Override
28     public void onClick(View view) {
29
30         // 1. create an intent pass class name or intnet action name
31         Intent intent = new Intent("com.hmkcode.android.ANOTHER_ACTIVITY");
32
33         // 2. create person object
34         Person person = new Person();
35         person.setName(etName.getText().toString());
36         person.setAge(Integer.parseInt(etAge.getText().toString()));
37
38         // 3. put person in intent data
39         intent.putExtra("person", person);
40
41         // 4. start the activity
42         startActivity(intent);
43     }
44
45 }

```

#### ■ **src/com/hmkcode/android/AnotherActivity.java**

- **AnotherActivity.java** is the activity we want to start to receive the person object. It has one text view to display person object "Person.toString()" sent from the starter activity.
- We can access to the sent data by getting a reference to the sent intent using **getIntent()**;
- Once we get the intent we can extract the object by calling intent.**getSerializableExtra**("person").
- Down cast the extracted object to the required type "(Person)".

```

1  package com.hmkcode.android;
2
3  import android.app.Activity;
4  import android.content.Intent;
5  import android.os.Bundle;
6  import android.widget.TextView;
7
8  public class AnotherActivity extends Activity {
9
10     TextView tvPerson;
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         // TODO Auto-generated method stub
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_another);
17
18         // 1. get passed intent
19         Intent intent = getIntent();
20
21         // 2. get person object from intent
22         Person person = (Person) intent.getSerializableExtra("person");
23
24         // 3. get reference to person textView
25         tvPerson = (TextView) findViewById(R.id.tvPerson);
26
27         // 4. display name & age on textView
28         tvPerson.setText(person.toString());
29
30     }
31 }

```

## ( 4 ) Define Activities in the Manifest XML File

- All activity should be defined with in the manifest.xml file

### ■ AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.hmkcode.android"
4      android:versionCode="1"
5      android:versionName="1.0" >
6
7      <uses-sdk
8          android:minSdkVersion="8"
9          android:targetSdkVersion="17" />
10
11     <application
12         android:allowBackup="true"
13         android:icon="@drawable/ic_launcher"
14         android:label="@string/app_name"
15         android:theme="@style/AppTheme" >
16
17         <activity
18             android:name="com.hmkcode.android.MainActivity"
19             android:label="@string/app_name" >
20             <intent-filter>
21                 <action android:name="android.intent.action.MAIN" />
22
23                 <category android:name="android.intent.category.LAUNCHER" />
24             </intent-filter>
25         </activity>
26
27         <activity
28             android:name="com.hmkcode.android.AnotherActivity"
29             android:label="@string/app_name" >
30             <intent-filter>
31                 <action android:name="com.hmkcode.android.ANOTHER_ACTIVITY" />
32                 <category android:name="android.intent.category.DEFAULT" />
33             </intent-filter>
34         </activity>
35     </application>
36
37 </manifest>
```

This entry was posted in Android and tagged activity, android on September 18, 2013

[<http://hmkcode.com/android-passing-java-object-another-activity/>].

---