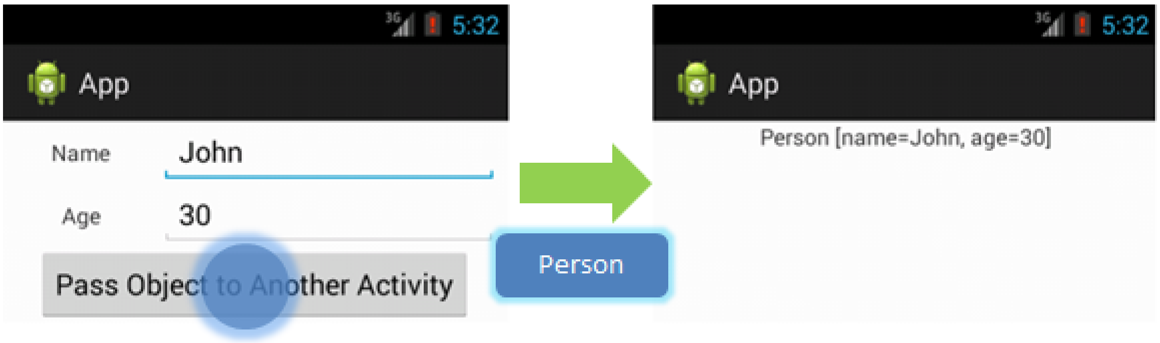
Android – Passing Java Object to Another Activity



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 activites.

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**
- Pakcage: **com.hmkcode.android**
- Keep other defualt 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.

```
package com.hmkcode.android;
```

```java

import java.io.Serializable;

public class Person implements Serializable{

    private static final long serialVersionUID = 1L;

    private String name;
    private int age;

        // getters & setters....

    @Override
    public String toString() {
        return "Person [name=" + name + ", age=" + age + "]";
    }
}
```

(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.

```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"
    tools:context=".MainActivity" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/tvName"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:gravity="center_horizontal"
            android:text="Name" />

        <EditText
            android:id="@+id/etName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:ems="10" >
            <requestFocus />
        </EditText>
    </LinearLayout>

    <LinearLayout
         android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
    <TextView
        android:id="@+id/tvAge"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center_horizontal"
        android:text="Age" />
    <EditText
        android:id="@+id/etAge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10" />
    </LinearLayout>

    <Button
        android:id="@+id/btnPassObject"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Pass Object to Another Activity" />

</LinearLayout>
```

- **res/layout/activity_another.xml**

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

```
<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"
     >

    <TextView
        android:id="@+id/tvPerson"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:layout_gravity="center"
        android:gravity="center_horizontal"
     />

</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.

```
package com.hmkcode.android;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener {

    Button btnPassObject;
    EditText etName, etAge;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnPassObject = (Button) findViewById(R.id.btnPassObject);
        etName = (EditText) findViewById(R.id.etName);
        etAge = (EditText) findViewById(R.id.etAge);

        btnPassObject.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {

        // 1. create an intent pass class name or intnet action name
        Intent intent = new Intent("com.hmkcode.android.ANOTHER_ACTIVITY");

        // 2. create person object
        Person person = new Person();
        person.setName(etName.getText().toString());
        person.setAge(Integer.parseInt(etAge.getText().toString()));

        // 3. put person in intent data
        intent.putExtra("person", person);

        // 4. start the activity
        startActivity(intent);
    }

}
```

- **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)”.

```
package com.hmkcode.android;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

public class AnotherActivity extends Activity {

    TextView tvPerson;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_another);

        // 1. get passed intent
        Intent intent = getIntent();

        // 2. get person object from intent
        Person person = (Person) intent.getSerializableExtra("person");

        // 3. get reference to person textView
        tvPerson = (TextView) findViewById(R.id.tvPerson);

        // 4. display name & age on textView
        tvPerson.setText(person.toString());

    }
}
```

(4) Define Activities in the Manifest XML File

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

AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.hmkcode.android"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name="com.hmkcode.android.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name="com.hmkcode.android.AnotherActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="com.hmkcode.android.ANOTHER_ACTIVITY" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

This entry was posted in Android and tagged activity, android on September 18, 2013 [http://hmkcode.com/android-passing-java-object-another-activity/] .