# Fragments Cont.

In this lecture we will see:

* How a fragment communicates with another fragment

Often you will want one Fragment to communicate with another, for example to change the content based on a user event. Basically, for two fragments to communicate, you need to make a Fragment A reference to the Fragment B and everything is resolved. However, as a good practice is the reuse of Fragment, it starts to not work because it will be with a high coupling and its reusability would be very low. The main idea is Fragment possible reuse in another Activity for example. This communication problem can be solved by using an Event Carrier, which is an interface that contains any method which will be responsible for loading the event of a Fragment to the other through the Activity.

**Define an Interface**

To allow a Fragment to communicate up to its Activity, you can define an interface in the Fragment class and implement it within the Activity. The Fragment captures the interface implementation during its onAttach() lifecycle method and can then call the Interface methods in order to communicate with the Activity.

**public interface** Communicator {

**public void** respond(**int** id);

}

This is the Event Carrier interface and will be implemented in the Activity

Considering the example from the last lesson, we have the implementation of the interface

**. . .**   
**public class** MainActivity **extends** AppCompatActivity **implements** Communicator {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 }  
 @Override **public void** respond(**int** data) {  
 FragmentManager manager = getFragmentManager();  
 Fragment2 frag = (Fragment2)manager.findFragmentById(R.id.***fragment2***);  
 frag.change(data);  
 }  
}

Notice that the *Fragment1* has access through the communicator attribute to the method implemented in MainActivity. The onAttach method, present in Fragment lifecycle was used to receive the operating system a reference of the Activity, and how it implements the communicator interface.

**public class** Fragment1 **extends** Fragment **implements** AdapterView.OnItemClickListener{

**private** ListView **list**;  
 **private** Communicator **comm**;  
 String [] **myarray**={. . .};

**public void** onAttach(Activity activity) {  
 **super**.onAttach(activity);  
 **comm** = (Communicator)activity;

}

. . .

}

There is no problem in making a cast to assign it to object.

**The code of Fragment1**

**public class** Fragment1 **extends** Fragment **implements** AdapterView.OnItemClickListener{  
 **private** ListView **list**;  
 **private** Communicator **comm**;  
 **public** String []**myarray**={**"Lecture1"**,**"Lecture2"**,**"Lecture3"**,**"Lecture4"**,**"Lecture5"**}

**public void** onAttach(Activity activity) {  
 **super**.onAttach(activity);  
 **comm** = (Communicator)activity; }  
 @Nullable  
 @Override  
 **public** View onCreateView  
 (LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 **return** inflater.inflate(R.layout.***activity\_fragment1***, container,

**false**); }  
 @Override  
 **public void** onActivityCreated(Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 **list** = (ListView) getActivity().findViewById(R.id.***list***);  
 ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(  
 getContext(), android.R.layout.***simple\_list\_item\_1***,**myarray**);  
 **list**.setAdapter(adapter);  
  
 **list**.setOnItemClickListener(**this**); }  
 @Override  
 **public void** onItemClick(AdapterView<?> parent,  
 View view, **int** position, **long** id) {  
 **comm**.respond(position);  
 }  
}

Fragment1 responsible for displaying the list on the screen and send the data to the Fragment2. Note that the Comunicator is set at the beginning of the class as a private attribute, it will be used only within that class.  
When you create a view "public View onCreateView (LayoutInflater inflater, ViewGroup container, savedInstanceState Bundle) {" return is a type view, so you must make a inflate, which is a method contained within the LayoutInflater class, responsible for "inflating "the layout XML. In other words, it transforms the XML into a type View object.

**onActivityCreated method**

**public void** onActivityCreated(Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 **list** = (ListView) getActivity().findViewById(R.id.***list***);  
 ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(  
 getContext(), android.R.layout.***simple\_list\_item\_1***,**myarray**);  
 **list**.setAdapter(adapter);  
  
 **list**.setOnItemClickListener(**this**);

}

it is recommended that the Fragment of UI operations are performed from the onActivityCreated method, because it ensures that the Activity has been completely created.

Within a ListView object type must attach the element that represents the list in the Fragment layout XML, for it is used findViewById method passing reference to that element. Thus, a ArrayAdapter type of object should be created to attach to this list an array of strings.

**onItemClick method**

@Override  
**public void** onItemClick(AdapterView<?> parent,  
 View view, **int** position, **long** id) {  
 **comm**.respond(position);  
}

the listener identifies the position that was clicked and pass as a parameter in the method respond. Remember that this method is defined in the Activity. Thus, the Fragment1 class is complete.

import android.app.Activity;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.ListView;

public class Fragment1 extends Fragment implements AdapterView.OnItemClickListener{

private ListView list;

private Communicator comm;

public String []myarray={"Lecture1","Lecture2","Lecture3","Lecture4","Lecture5"};

public void onAttach(Activity activity) {

super.onAttach(activity);

comm = (Communicator)activity;

}

@Nullable

@Override

public View onCreateView

(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

return inflater.inflate(R.layout.activity\_fragment1, container, false); }

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

list = (ListView) getActivity().findViewById(R.id.list);

ArrayAdapter<String> adapter = new ArrayAdapter<String>(

getContext(), android.R.layout.simple\_list\_item\_1,myarray );

list.setAdapter(adapter);

list.setOnItemClickListener(this); }

@Override

public void onItemClick(AdapterView<?> parent,

View view, int position, long id) {

comm.respond(position);

}

}

Another variant of the Fragment1 code is

**import** android.app.Activity;  
**import** android.os.Bundle;  
**import** android.support.annotation.Nullable;  
**import** android.support.v4.app.Fragment;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.AdapterView;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.ListView;  
  
**public class** Fragment1 **extends** Fragment{  
 **private** ListView **list**;  
 **private** Communicator **comm**;  
 **public** String []**myarray**

={**"Lecture1"**,**"Lecture2"**,**"Lecture3"**,**"Lecture4"**,**"Lecture5"**};  
  
 **public void** onAttach(Activity activity) {  
 **super**.onAttach(activity);  
 **comm** = (Communicator)activity;  
  
 }  
 @Nullable  
 @Override  
 **public** View onCreateView  
 (LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 **return** inflater.inflate(R.layout.***activity\_fragment1***, container,

**false**); }  
 @Override  
 **public void** onActivityCreated(Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 **list** = (ListView) getActivity().findViewById(R.id.***list***);  
 ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(  
 getContext(), android.R.layout.***simple\_list\_item\_1***,**myarray** );  
 **list**.setAdapter(adapter);  
  
 *//list.setOnItemClickListener(this);* **list**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 **public void** onItemClick(AdapterView<?> parent,View view,

**int** position, **long** id) {  
 listItemClicked(view,position);  
 }  
 });  
  
  
 }  
 **void** listItemClicked(View v, **int** position){  
 **comm**.respond(position);  
  
 }  
}

**Fragment2 code  
import** android.app.Activity;  
**import** android.app.Fragment;  
**import** android.content.res.Resources;  
**import** android.os.Bundle;  
**import** android.support.annotation.Nullable;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.TextView;  
  
**public class** Fragment2 **extends** Fragment {  
 **private** TextView **text**;  
 **private** Activity **activity**;  
  
 **public void** onAttach(Activity activity) {  
 **super**.onAttach(activity);  
 **this**.**activity** = activity;  
 }  
  
 @Nullable  
 @Override  
 **public** View onCreateView(LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
 **return** inflater.inflate(R.layout.***activity\_fragment2***,  
 container, **false**);  
 }  
  
 @Override  
 **public void** onActivityCreated(Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 **text** = (TextView) **activity**.findViewById(R.id.***textView1***);  
 }  
  
 **public void** change(**int** data) {  
 Resources res = getResources();  
 String[] desc = res.getStringArray(R.array.***descrip***);  
 **text**.setText(desc[data]);  
 }  
}

The Fragment2 has some very similar methods Fragment1: the onAttach copy a reference to the Activity that will be used in onActivityCreated method and onCreateView method does the same thing in Fragment1A.

**public void** onAttach(Activity activity) {  
 **super**.onAttach(activity);  
 **this**.**activity** = activity;  
}

The onActivityCreated method also has the same function as the Fragment1, with the exception here will be the link between a TextView object responsible for displaying the elements on the screen. Note that, for didactic purposes, here was made the reference to the type of object Activity instantiated earlier in onAttach to simplify. The onAttach method could be deleted and the object Activity type could be replaced by getActivity() method.

@Override  
**public void** onActivityCreated(Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 **text** = (TextView) **activity**.findViewById(R.id.***textView1***);  
}

**Method change**

method responsible for displaying on the screen the contents of the array whose content was sent by the fragmenting. Here again, the resources are loaded by getResources() method, the array of strings containing data the mass regarding the description is stored in an Array object of type String, and finally, the TextView object type displays the data in the reference screen the position of the array sent by the fragmenting.

**public void** change(**int** data) {  
 Resources res = getResources();  
 String[] desc = res.getStringArray(R.array.***description***);  
 **text**.setText(desc[data]);  
}

So the complete code of the second fragment is:

**import** android.app.Activity;  
**import** android.app.Fragment;  
**import** android.content.res.Resources;  
**import** android.os.Bundle;  
**import** android.support.annotation.Nullable;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.TextView;  
  
**public class** Fragment2 **extends** Fragment {  
 **private** TextView **text**;  
 **private** Activity **activity**;  
  
 **public void** onAttach(Activity activity) {  
 **super**.onAttach(activity);  
 **this**.**activity** = activity;  
 }  
  
 @Nullable  
 @Override  
 **public** View onCreateView(LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
 **return** inflater.inflate(R.layout.***activity\_fragment2***,  
 container, **false**);  
 }  
  
 @Override  
 **public void** onActivityCreated(Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 **text** = (TextView) **activity**.findViewById(R.id.***textView1***);  
 }  
  
 **public void** change(**int** data) {  
 Resources res = getResources();  
 String[] desc = res.getStringArray(R.array.***description***);  
 **text**.setText(desc[data]);  
 }  
}

**OR**

**import** android.app.Activity;  
**import** android.app.Fragment;  
**import** android.content.res.Resources;  
**import** android.os.Bundle;  
**import** android.support.annotation.Nullable;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.TextView;  
  
**public class** Fragment2 **extends** Fragment {  
 **private** TextView **text**;  
 **private** Activity **activity**;  
  
 @Nullable  
 @Override  
 **public** View onCreateView(LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState)

{  
 **return** inflater.inflate(R.layout.***activity\_fragment2***,  
 container, **false**);  
 }  
  
 @Override  
 **public void** onActivityCreated(Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 **text** = (TextView)getActivity().findViewById(R.id.***textView1***);  
 }  
  
 **public void** change(**int** data) {  
 Resources res = getResources();  
 String[] desc = res.getStringArray(R.array.***description***);  
 **text**.setText(desc[data]);  
 }  
}

**Testing**

# 

# An other variant of the

# Class Work

Create a project with the basic activity option. The Project has an Activity and two fragments. Use the following files and complete the code.

The main activity layout includes:

<**fragment  
 android:id="@+id/fragment1"  
 android:name=". . . . Fragment1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 tools:layout="@layout/fragment1"** />  
  
<**fragment  
 android:id="@+id/fragment2"  
 android:name=". . .. Fragment2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_centerVertical="true"  
 tools:layout="@layout/fragment2"** />

Create an interface with the method onButtonClick(**int** font, String text)   
This method with be implemented in MainActivity.  
  
 }

MainActivity.java

**public class** MainActivity **extends** Activity **. .** . . . {  
 **private int fontsize**=10;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);

//set the toolbar  
  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.***main\_menu***,menu);  
 **return true**;  
 }  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** R.id.***menuitem\_fragment1***:  
 **fontsize**=10;  
 Toast.*makeText*(**this**, **"Font 10"**,  
 Toast.***LENGTH\_SHORT***).show();  
 **return true**;  
 **case** R.id.***menuitem\_fragment2***:  
 **fontsize**=20;  
 Toast.*makeText*(**this**, **"Font 50"**,  
 Toast.***LENGTH\_SHORT***).show();  
 **return true**;

. . .

}  
  
 **return false**;  
 }  
  
 **public void** onButtonClick(**int** font, String text){  
 . . . .  
  
 }  
}

**Fragment 1 Layout**

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**EditText  
 android:id="@+id/editText1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="17dp"  
 android:layout\_centerHorizontal="true"  
 android:ems="10"  
 android:inputType="text"** >  
 <**requestFocus** />  
 </**EditText**>  
 <**Button  
 android:id="@+id/button1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/editText1"  
 android:text="Change Text"** />  
  
  
</**RelativeLayout**>

Fragment1.java

**import** android.app.Activity;  
**import** android.app.Fragment;  
**import** android.os.Bundle;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
  
*/\*\*  
 \* Created by tamar on 11/7/2017.  
 \*/***public class** Fragment1 **extends** Fragment {  
  
 **private static** EditText *edittext*;  
 **private int fontsize**=10;  
 interface. . **activityCallback**;  
  
 @Override  
 **public void** onAttach(Activity activity){  
 **super**.onAttach(activity);  
 **try** {  
 **activityCallback**=(interface. . .) activity;  
 }**catch** (ClassCastException e){  
 **throw new** ClassCastException(

activity.toString()+ **"must implement the listener"**);  
 }  
  
 }  
 @Override  
 **public** View onCreateView(LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
 *// Inflate the layout for this fragment* View view = inflater.inflate(R.layout. ***. . )***,  
 container, **false**);  
  
 *edittext* = (EditText) view.findViewById(R.id.***editText1***);  
  
 **final** Button button =  
 (Button) view.findViewById(R.id.***button1***);  
 button.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 buttonClicked(v);  
  
 }  
 });  
  
 **return** view;  
 }  
 **void** buttonClicked(View v){  
 **. . .**   
 }  
}

**Fragment 2 Layout**

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:text="Fragment Two"  
 android:textAppearance="?android:attr/textAppearanceLarge"** />  
  
</**RelativeLayout**>

Fragment2.java

**public class** TextFragment **extends** Fragment {  
 **private static** TextView *textview*;  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.***text\_fragment***,  
 container, **false**);  
 *textview*=(TextView)view.findViewById(R.id.***textView1***);  
  
 **return** view;  
 }  
 **public void** changeTextProperties(**int** fontsize, String text){  
 *. . .*

}  
}

## References:

<http://mrbool.com/communication-between-android-fragments/34435>

<https://developer.android.com/training/basics/fragments/communicating.html>

http://www.techotopia.com/index.php/Using\_Fragments\_in\_Android\_-\_A\_Worked\_Example