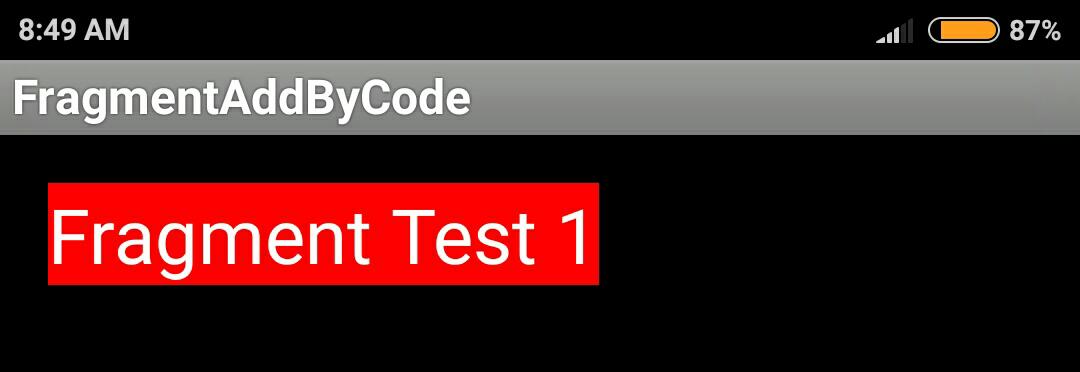
Fragment – Mảnh

Fragment là 1 mảnh, mảng, miếng của Interface, activity. Có hàng đợi (backstack) chứ để quản lý. Thêm, xoá, thay đổi dễ dàng.

# Add Fragment

## Add by Code

**Bước 1: Tạo Fragment**

XML file: **res/layout/fragment\_test\_1.xml**

<LinearLayout…>

<TextView…

android:text="Fragment Test 1"

android:id="@+id/textView"/>

</LinearLayout>

Java file: **Test1Fragment.java**

public class Test1Fragment extends Fragment {

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_test\_1, container, false);

}

}

**Bước 2: Gọi vào trong Activities:**

**layout/activity\_main.xml**

<RelativeLayout…

tools:context=".MainActivity"

android:id="@+id/myLayout">

</RelativeLayout>

**MainActivity.java**

protected void onCreate(Bundle savedInstanceState) {

...

// add fragment by code

Test1Fragment fragment = new Test1Fragment();

FragmentManager manager = getFragmentManager();

FragmentTransaction transaction = manager.beginTransaction();

transaction.add(R.id.myLayout, fragment, "fragment\_tag");

transaction.commit();

}

## Add fragment by XML

Trong file, tạo fragment như trên, MainActivity.xml chỉ cần gọi:

<RelativeLayout…>

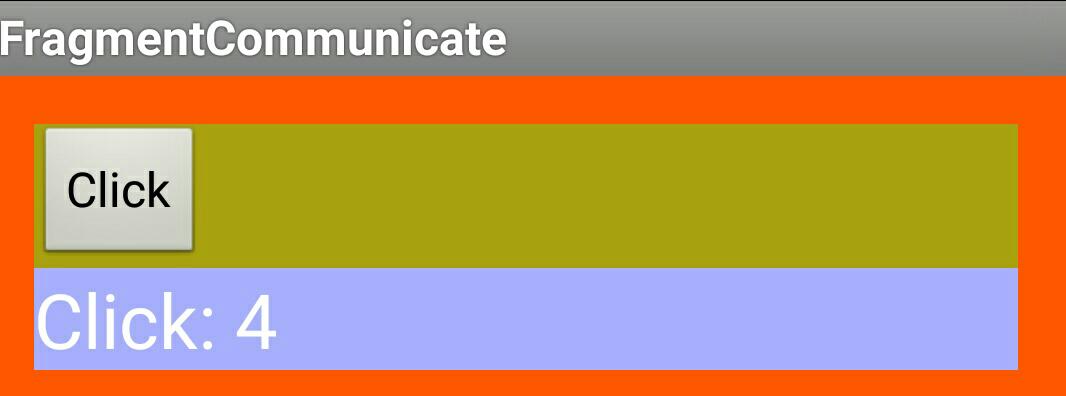
<fragment…

android:name="com.imark.nghia.fragmentaddbyxml.Test1Fragment"

tools:layout="@layout/fragment\_test\_1"/>

</RelativeLayout>

# Communicate – Giao tiếp với Fragment



Click Button trong FragmentA (xanh lục) sẽ truyền số lần vào FragmentB (tím)

**Bước 1: tạo interface giao tiếp**

public interface IFragmentCommunicate {

/\*\* phản hồi, chuyển dữ liệu giũa các fragment \*/

public void respond(String value);

}

**Bước 2: Tạo FragmentA**

**fragment\_a.xml**

<LinearLayout…>

<Button…

android:id="@+id/btnClick"

android:text="Click"/>

</LinearLayout>

**FragmentA**.**java**

public class FragmentA extends Fragment {

private static final String EXTRA\_NAME\_COUNTER = "COUNTER";

private Button btnClick;

private int mCounter; // lưu số lần nhấn

private IFragmentCommunicate fragmentCommunicate; // giao tiếp với mainActivity

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// load lại trạng thái của FRAGMENT khi có thay đổi (Example: xoay màn hình)

if (savedInstanceState == null) {

// nếu không có giá trị

mCounter = 0;

} else {

// có giá trị

mCounter = savedInstanceState.getInt(EXTRA\_NAME\_COUNTER, 0);

}

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_a, container, false);

}

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

// mainActivity trung gian, chuyển dữ liệu

fragmentCommunicate = (IFragmentCommunicate)getActivity();

btnClick = (Button)getActivity().findViewById(R.id.btnClick);

btnClick.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

btnClick\_onClick(view);

}

});

}

@Override

public void onSaveInstanceState(Bundle outState) {

// lưu trạng thái hiện tại của FRAGMENT

// khi xoay màn hình, fragment sẽ load lại, trở về [onCreate]

super.onSaveInstanceState(outState);

// ----- Lưu trạng thái khi có thay đổi -----

outState.putInt(EXTRA\_NAME\_COUNTER, mCounter);

}

private void btnClick\_onClick(View view) {

mCounter++;

// gởi mainActivity respond, respond này sẽ chuyển dữ liệu cho fragmentB

fragmentCommunicate.respond("Click: " + mCounter);

}

}

**Bước 3: tạo Fragment B**

**File: fragment\_b.xml**

<LinearLayout…>

<TextView… android:id="@+id/lblText"/>

</LinearLayout>

**File: FragmentB.java**

public class FragmentB extends Fragment {

private static final String EXTRA\_NAME\_DATA = "data";

private TextView lblText;

private String mData;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_b, container, false);

return view;

}

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

lblText = (TextView)getActivity().findViewById(R.id.lblText);

// có giao tiếp views nên dùng [onCreateView]

//---- Update data ----

if (savedInstanceState == null) {

mData = "";

}

else {

mData = savedInstanceState.getString(EXTRA\_NAME\_DATA);

lblText.setText(mData);

}

}

@Override

public void onSaveInstanceState(Bundle outState) {

super.onSaveInstanceState(outState);

// ----- Lưu trạng thái ----

outState.putString(EXTRA\_NAME\_DATA, mData);

}

public void changeText(String data){

mData = data;

lblText.setText(mData);

}

}

**Bước 4: gọi 2 Fragment trong Activity**

**File: main\_activity.xml**

<LinearLayout…>

<fragment…

android:id="@+id/fragment\_1"

android:name="com.imark.nghia.fragmentcommunicate.FragmentA"

tools:layout="@layout/fragment\_a" />

<fragment…

android:id="@+id/fragment\_2"

android:name="com.imark.nghia.fragmentcommunicate.FragmentB"

tools:layout="@layout/fragment\_b" />

</LinearLayout>

**MainActivity.java**

public class MainActivity extends Activity implements IFragmentCommunicate {

//...

/\*\*

\* phản hồi, chuyển dũ liệu giũa các fragment

\* Explain:

\* FragmentA chuyển MainActivity thành IFragmentCommunicate

\* FragmentA gọi hàm respone() trong MainActivity

\* Hàm respone() trong MainActivity: tìm FragmentB

\* MainActivity gọi hàm changeText() trong FragmentB

\*/

@Override

public void respond(String value) {

// tìm fragment b

FragmentManager fragmentManager = getFragmentManager();

FragmentB fragmentB = (FragmentB) fragmentManager.findFragmentById(R.id.fragment\_2);

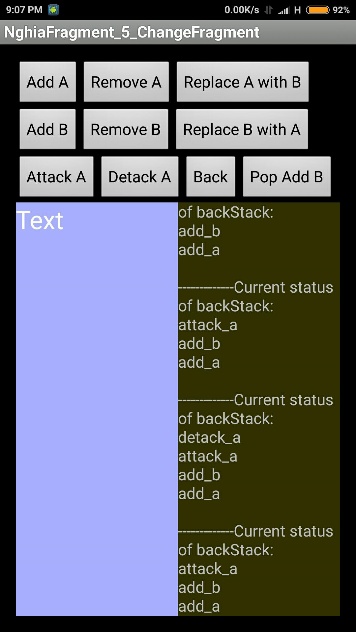
// đẩy giá trị qua FRAGMENT B

fragmentB.changeText(value);

}

}

# Xủ lý Fragment trên Activity



public class MainActivity extends Activity {

private FragmentManager fragManager;

private TextView txtMessage;

@Override

protected void onCreate(Bundle savedInstanceState) {…

// hiện backstack mỗi lần thay đổi

fragManager.addOnBackStackChangedListener(new FragmentManager.OnBackStackChangedListener() {

@Override

public void onBackStackChanged() {

fragManager\_onBackStackChanged();

}

});

}

public void addA(View view) {

FragmentA f1 = new FragmentA();

FragmentTransaction transaction = fragManager.beginTransaction();

transaction.add(R.id.layoutNghia, f1, "A");

transaction.addToBackStack("add\_a"); // save backstack, back sẽ remove trong Back Stack

transaction.commit(); // lưu thay đổi

}

public void removeA(View view) {

FragmentA f1 = (FragmentA) fragManager.findFragmentByTag("A");

if (f1 == null) {

Toast.makeText(this, "Not found", Toast.LENGTH\_LONG).show();

} else {

FragmentTransaction transaction = fragManager.beginTransaction();

transaction.remove(f1);

transaction.addToBackStack("remove\_a");

transaction.commit();

}

}

public void replaceAwithB(View view) {

FragmentB f2 = new FragmentB();

FragmentTransaction transaction = fragManager.beginTransaction();

transaction.replace(R.id.layoutNghia, f2, "B");

transaction.addToBackStack("replace\_a\_with\_b");

transaction.commit();

}

public void addB(View view) {

FragmentB f2 = new FragmentB();

FragmentTransaction transaction = fragManager.beginTransaction();

transaction.add(R.id.layoutNghia, f2, "B");

transaction.addToBackStack("add\_b");

transaction.commit(); // lưu thay đổi

}

public void removeB(View view) {

FragmentB f2 = (FragmentB) fragManager.findFragmentByTag("B");

if (f2 == null) {

Toast.makeText(this, "Not found", Toast.LENGTH\_LONG).show();

} else {

FragmentTransaction transaction = fragManager.beginTransaction();

transaction.remove(f2);

transaction.addToBackStack("remove\_b");

transaction.commit(); // lưu thay đổi

}

}

public void replaceBwithA(View view) {

FragmentA f1 = new FragmentA();

FragmentTransaction transaction = fragManager.beginTransaction();

transaction.replace(R.id.layoutNghia, f1, "A");

transaction.addToBackStack("replace\_b\_width\_a");

transaction.commit(); // lưu thay đổi

}

public void attackA(View view) {

FragmentA f1 = (FragmentA) fragManager.findFragmentByTag("A");

if (f1 == null) {

Toast.makeText(this, "Not found", Toast.LENGTH\_LONG).show();

} else {

FragmentTransaction transaction = fragManager.beginTransaction();

transaction.attach(f1);

transaction.addToBackStack("attack\_a");

transaction.commit(); // lưu thay đổi

}

}

public void detackA(View view) {

FragmentA f1 = (FragmentA) fragManager.findFragmentByTag("A");

if (f1 == null) {

Toast.makeText(this, "Not found", Toast.LENGTH\_LONG).show();

} else {

FragmentTransaction transaction = fragManager.beginTransaction();

transaction.detach(f1);

transaction.addToBackStack("detack\_a");

transaction.commit(); // lưu thay đổi

}

}

public void back(View view) {

// xóa hết entry cuối cùng trong backstack

fragManager.popBackStack();

}

public void popAddB(View view) {

// xóa hết entry add\_b trong backStack

fragManager.popBackStack("add\_b", FragmentManager.POP\_BACK\_STACK\_INCLUSIVE);

}

private void fragManager\_onBackStackChanged() {

// hiểu thị khi thay đổi BackStack

String message = txtMessage.getText().toString();

message += "\n\n-------------";

message += "Current status of backStack: ";

// hiện tên entry(mục) trong backstack(ngăn xếp)

int count = fragManager.getBackStackEntryCount();

for (int i = count-1; i>=0; i--) {

FragmentManager.BackStackEntry entry = fragManager.getBackStackEntryAt(i);

message += "\n" + entry.getName();

}

txtMessage.setText(message);

}

}