FREE eBook

LEARNING android-fragments

Free unaffiliated eBook created from
Stack Overflow contributors.

#android-fragments

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: android-fragments

It is an unofficial and free android-fragments ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official android-fragments.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with android-fragments

Remarks

Fragments are very important components of user interface in android apps. They were introduced first in Android 3.0 (Honeycomb) API.

Understanding design paradigm of Fragments

Fragments were introduced for primarily supporting modular and flexible UI on large screen devices such as tablets.

Fragments are managed by an activity. Usually each fragment represents a portion of a screen. There can be more than one fragment in an activity. Fragments may also be called *sub-activities*. When you add a fragment as a part of your activity layout, it lives in a ViewGroup inside the activity's view hierarchy and the fragment defines its own view layout.

LIFECYCLE

Just like an activities, fragments also have a lifecycle. A fragment gets notified for following events.

1. Get attached to activity - **onAttach(Activity)**
2. Create fragment - **onCreate(Bundle)**
3. Create View - **onCreateView(LayoutInflater, ViewGroup, Bundle)**
4. Activity creation - **onActivityCreated(Bundle)**
5. View state restored - **onViewStateRestored(Bundle)**
6. Made visible to user - **onStart()**
7. start of user interaction - **onResume()**
8. pause of user interaction - **onPause()**
9. Made invisible to user - **onStop()**
10. On view destruction - **onDestroyView()**
11. Destroy fragment - **onDestroy()**
12. Get detached from an activity - **onDetach()**

As a programmer, you should override various lifecycle callback methods, typically we implement onCreate(), onCreateView() and onPause() methods.

Subclasses of Fragment

1. DialogFragment - For displaying the floating dialog
2. ListFragment - For displaying list of items
3. PreferenceFragment - Useful for creating settings activity

---

References

1. https://developer.android.com/guide/components/fragments.html
2. https://developer.android.com/reference/android/app/Fragment.html

Examples

Adding Fragments

Adding a Fragment Statically

File: *activity_main.xml*

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <fragment
        android:id="@+id/fragment2"
        android:name="com.example.fragmentexample.Fragment2"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="1"
        />

    <fragment
        android:id="@+id/fragment1"
        android:name="com.example.fragmentexample.Fragment1"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="1"
         />

</LinearLayout>
```

File: *fragment1.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#00ff00"
     >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="fragment frist"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</LinearLayout>
```

File: *fragment2.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#0000ff"
     >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Fragment"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</LinearLayout>
```

File: *MainActivity.java*

```
package com.example.fragmentexample;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

File: *Fragment1.java*

```
package com.example.fragmentexample;

import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class Fragment1 extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        return inflater.inflate(R.layout.fragment1,container, false);
    }

}
```

File: *Fragment2.java*

```
package com.example.fragmentexample;
```

```
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class Fragment2 extends Fragment {

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        return inflater.inflate(R.layout.fragment2,container, false);
    }

}
```

Adding a Fragment Dynamically

activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <FrameLayout
        android:id="@+id/container1"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="1"
        />

    <FrameLayout
        android:id="@+id/container2"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="1"
         />

</LinearLayout>
```

FrameLayout is acting as fragment container.

MainActivity class

File: MainActivity.java

```
package com.example.fragmentexample;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        loadFragment(this, R.id.container1,new Fragment1(),"fragment1");
        loadFragment(this, R.id.container2,new Fragment2(),"fragment2");
    }

    public static void loadFragment(Activity activity, int containerId, Fragment fragment,
String tag)
    {
        activity.getSupportFragmentManager().beginTransaction().
            replace(containerId, fragment,tag).commitAllowingStateLoss();
    }
}
```

Read Getting started with android-fragments online: https://riptutorial.com/android-fragments/topic/1933/getting-started-with-android-fragments

Credits

S. No	Chapters	Contributors
1	Getting started with android-fragments	chandil03, Community, Daniel Nugent, Laurel, Nicolai Weitkemper, Rubin Nellikunnathu, shahharshil46