



បណ្ណបណ្ណាលក្នុង សម្បត្តិ អេច អ ឌី

Korea Software HRD Center

Android Fragments

ណែនាំដោយ : Dr. Kim Tae Kyung



<http://www.kshrd.com.kh>

មាតិកា

1. Overview
2. Fragment Life Cycle
3. ការបង្កើត Fragment
4. ការ Add Fragment ទៅកាន់ Activity នៅពេល Runtime
5. ការ Replace Fragment នៅក្នុង Activity
6. ការទំនាក់ទំនងទៅកាន់ Activity & Fragment
7. ការទំនាក់ទំនងរវាង Fragment & Fragment
8. Deliver a Message to a Fragment

1. Overview

អ្វីទៅជា Fragment?

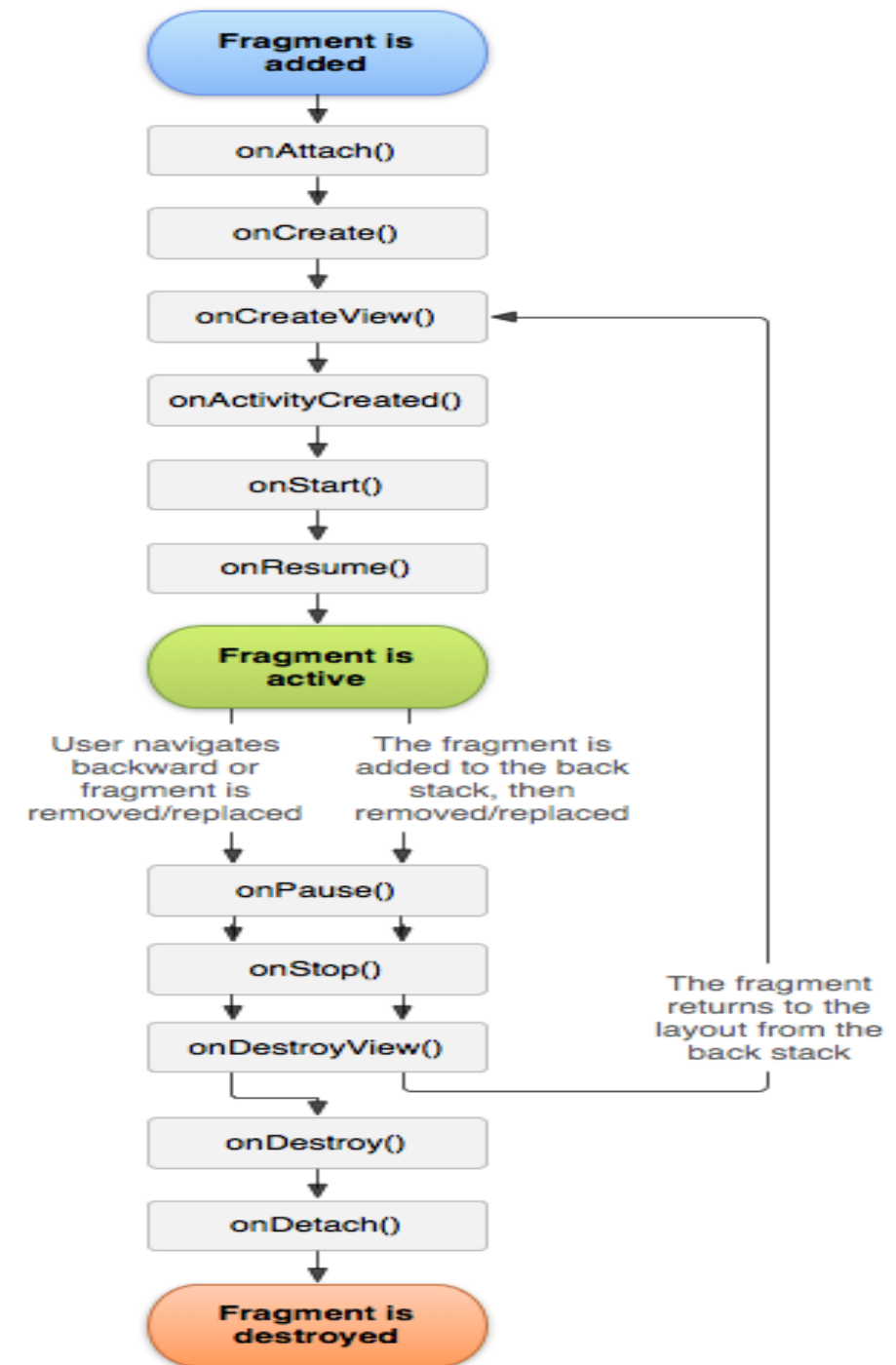
- **Android Fragment** ជាផ្នែកមួយនៃ activity ឬហៅថា sub-activity ។
- ការ configure fragments ក្នុង activity layout អាចប្រែប្រួលទៅតាមទំហំ screen (a small screen might show one fragment at a time, but a large screen can show two or more)



1. Overview (ត)

- លក្ខណៈសំខាន់របស់ **Fragment** មានដូចជា៖
- វាមាន **layout** និង **behavior** ជាមួយនិង **life cycle callbacks** របស់វាផ្ទាល់
- យើងអាចធ្វើការ **add** និង **remove fragments** នៅក្នុង activity អំឡុងពេលកំពុង run
- យើងអាច **combine multiple fragments** នៅក្នុង Activity តែមួយ
- Fragment មួយអាចត្រូវបានប្រើក្នុង **activity** ច្រើន
- Life cycle របស់ fragment គឺមានទំនាក់ទំនងទៅនឹង life cycle របស់ activity មេ មានន័យថា ពេលដែល activity ត្រូវបានបិទ នោះ fragment ដែលនៅក្នុង activity នោះនឹងត្រូវបញ្ឈប់ដែរ
- Fragment ត្រូវបានដាក់បញ្ចូលទៅក្នុង Android API ក្នុង Honeycomb version (API version 11)

2. Fragment Life Cycle



2. Fragment Life Cycle (ត)

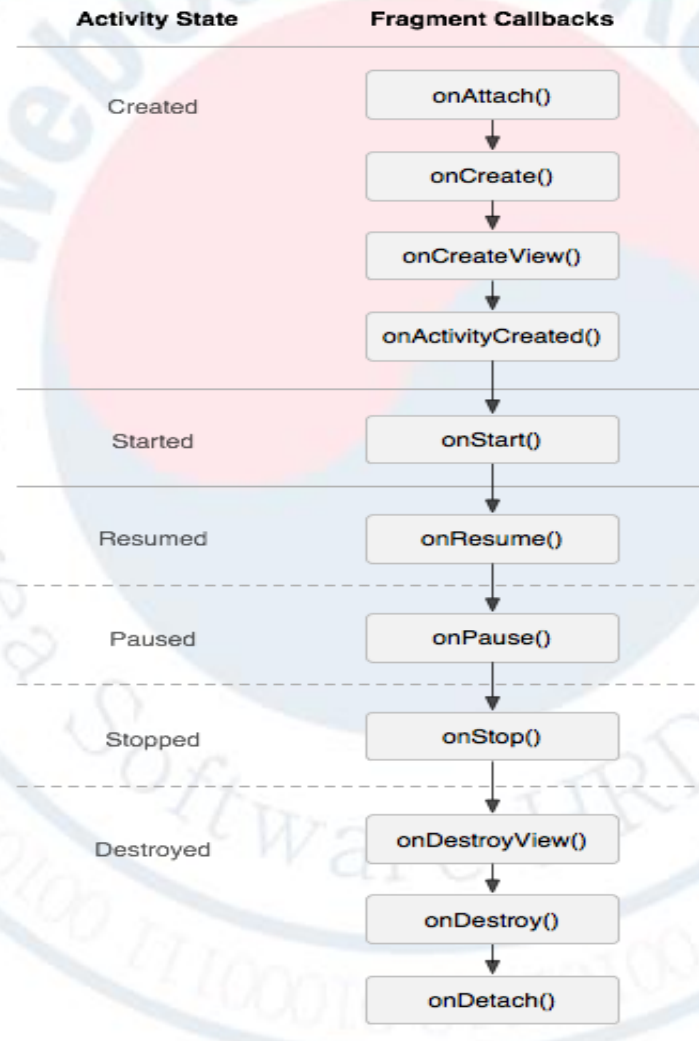
Methods មួយចំនួនដែលយើងអាច override ក្នុង fragment class ៖

Methods	Description
onAttach()	ត្រូវបានហៅបន្ទាប់ពី fragment បាន associated ជាមួយនឹង activity របស់វា
onCreate()	ត្រូវបានហៅនៅពេលបង្កើត fragment
onCreateView()	ត្រូវបានហៅនៅពេលដែល fragment draw User Interface របស់វាលើកដំបូង
onActivityCreated()	ត្រូវបានហៅនៅពេលដែល method onCreate() របស់ Activity បានបញ្ចប់
onStart()	ត្រូវបានហៅនៅពេលដែល fragment ត្រូវបានបង្ហាញ

2. Fragment Life Cycle (ត)

Methods	Description
onResume()	ត្រូវបានហៅនៅពេលដែល fragment កំពុង Active
onPause()	ត្រូវបានហៅនៅពេលដែល user ចាកចេញពី fragment
onStop()	ត្រូវបានហៅនៅពេលដែល method onStop() របស់ Activity ដំណើរការ
onDestroyView()	សម្រាប់ destroy fragment view
onDestroy()	សម្រាប់ clean up ចុងក្រោយគេ
onDetach()	ត្រូវបានហៅនៅពេលដែល fragment បានផ្តាច់ចេញពី Activity

2. Fragment Life Cycle (ᄇ)



3. របៀបបង្កើត Fragment

ដើម្បីបង្កើត fragments:

1. ត្រូវធ្វើការ **extend** ចេញពី **Fragment class**
2. អ្នកអាច **insert a fragment** ចូលទៅក្នុង **activity layout** ដោយ declare fragment ក្នុង activity layout file (**<fragment>** element)

3. ကနဦး Fragment (က)

1. Extend ကို Fragment class

File: Fragment.java

```
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.ViewGroup;

public class ArticleFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.article_view, container, false);
    }
}
```

3. ကယဉ္ဇေန် Fragment (က)

2. Add Fragment to Activity Using XML

File: news_articles.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="com.example.news.ArticleListFragment"
        android:id="@+id/list"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
    <fragment android:name="com.example.news.ArticleReaderFragment"
        android:id="@+id/viewer"
        android:layout_weight="2"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
</LinearLayout>
```

4. ការ Add Fragment ទៅកាន់ Activity នៅពេល Runtime

- ប្រសិនបើយើងមានបំណងចង់ផ្លាស់ប្តូរ Fragment នៅពេល Run Time យើងមិនគួរ កំណត់ Fragment នៅក្នុង Layout File នោះឡើយ។
- **FragmentManager** Class ត្រូវបានប្រើដើម្បីបង្កើត **FragmentTransaction** ដែលអាចអោយយើង ធ្វើការ add, remove, replace ហើយ perform fragment transactions ដទៃទៀត។
- ដើម្បី Add Fragment នៅពេល Run Time បាន យើងត្រូវបង្កើត View Container មួយ ដើម្បីដាក់ Fragment នោះ។

4. ការ Add Fragment ទៅកាន់ Activity នៅពេល Runtime (ត)

ឧទាហរណ៍៖ យើងមាន Layout File: **res/layout/news_articles.xml**

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/fragment_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

4. ការ Add Fragment ទៅកាន់ Activity នៅពេល Runtime (ត)

នៅក្នុង MainActivity យើងហៅ
`getFragmentManager()` ដើម្បី Get
`FragmentManager` ។ បន្ទាប់មកហៅ
`beginTransaction()` ដើម្បីបង្កើត
`FragmentTransaction` រួចហៅ
Method `add()` ដើម្បី Add Fragment ។
បន្ទាប់ យើងចាំបាច់ ហៅ Method
`commit()` ដើម្បីអោយវាដំណើរការ។

```
// Create fragment and pass position argument to the fragment
FragmentA newFrag = FragmentA.newInstance("Hello");
FragmentTransaction transaction =
getFragmentManager().beginTransaction();
// Add fragment into FrameLayout
transaction.add(R.id.fragment_container, newFrag);
// Commit the transaction
transaction.commit();
```


5. ការ Replace Fragment នៅក្នុង Activity

- ការ replace a fragment គឺស្រដៀងនឹងការ Add Fragment ដែរ គ្រាន់តែ change Method `add()` ទៅជា Method `replace()` វិញ។
- កាលណា យើងចង់ replace រឺ remove Fragment យើងគួរ Call `addToBackStack()` មុនពេល commit FragmentTransaction ដើម្បីអោយ User អាច navigate backward បាន។

5. ការ Replace Fragment នៅក្នុង Activity (ត)

ឧទាហរណ៍៖

```
// Create fragment and pass positin argment to the fragment
FragmentB newFrag = FragmentB.newInstance(position);
FragmentManager transaction = getFragmentManager().beginTransaction();
// Replace whatever is in the fragment_container view with this
// fragment and add the transaction to back stack, so user can navigate back
transaction.replace(R.id.fragment_container, newFrag);
transaction.addToBackStack(null);
// Commit the transaction
transaction.commit();
```

6. ការទំនាក់ទំនងទៅកាន់ Activity & Fragment

- អ្វីទៅជា ការទំនាក់ទំនងទៅកាន់ Activity ?
 - គឺជាដំណើរការនៃ Fragment ទៅកាន់ Activity តាមរយៈ `getActivity()` method និងការធ្វើការងារយ៉ាងងាយស្រួល ដូចជា ស្វែងរក View នៅក្នុង Activity Layout ។

ឧ: ការចាប់យក View ណាមួយ

```
View listView = getActivity().findViewById(R.id.list);
```

6. ការទំនាក់ទំនងទៅកាន់ Activity & Fragment (ត)

- ❑ Activity អាចហៅនូវ method នៅក្នុង Fragment ដោយទទួលបាន Reference ទៅកាន់ Fragment (From FragmentManager) និង អាចប្រើប្រាស់នូវ Method : `findFragmentById()` or `findFragmentByTag()`

ឧ: ការចាប់យក Fragment

```
ExampleFragment fragment = (ExampleFragment) getFragmentManager()  
    .findFragmentById(R.id.example_fragment);
```

7. ការទំនាក់ទំនងរវាង Fragment & Fragment

Fragment ពីរប្រើប្រាស់ អាចទំនាក់ទំនងគ្នាបានគឺតាមរយៈ Interface និង Activity ។ Interface នេះគឺត្រូវបង្កើតក្នុង Fragment ណាដែលចង់បោះតំលៃអ្វីមួយទៅអោយ Fragment ដទៃទៀត។ ចំណែក Activity ដែលផ្ទុកនូវ Fragment នេះត្រូវ implements Interface ដែលបានបង្កើតនេះ ។

```
public class FragmentA extends ListFragment {  
    ...  
    // Container Activity must implement this interface  
    public static interface OnArticleSelectedListener {  
        public void onArticleSelected(int position);  
    }  
    ...  
}
```


7. ការទំនាក់ទំនងរវាង Fragment & Fragment (ត)

ដើម្បីបញ្ជាក់អោយកាន់តែច្បាស់ថា Activity បាន implement ពី Interface ឬ អត់ !

```
public class FragmentA extends ListFragment {
    public static OnArticleSelectedListener mListener;
    ...
    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
        try {
            mListener = (OnArticleSelectedListener) activity;
        } catch (ClassCastException e) {
            throw new ClassCastException(activity.toString());
        }
    }
    ...
}
```


7. ករណីទាំងពីរទៀត Fragment & Fragment (ត)

ឧ: នៅពេលដែលចុច List Item ណាមួយ, Fragment ផ្សេងអាចចាប់យកបានតាមរយៈ Activity

```
public class FragmentA extends ListFragment {  
    public static OnArticleSelectedListener mListener;  
    ...  
    @Override  
    public void onListItemClick(ListView l, View v, int position, long id) {  
        // Send position to the host activity  
        mListener.onArticleSelected(position);  
    }  
    ...  
}
```

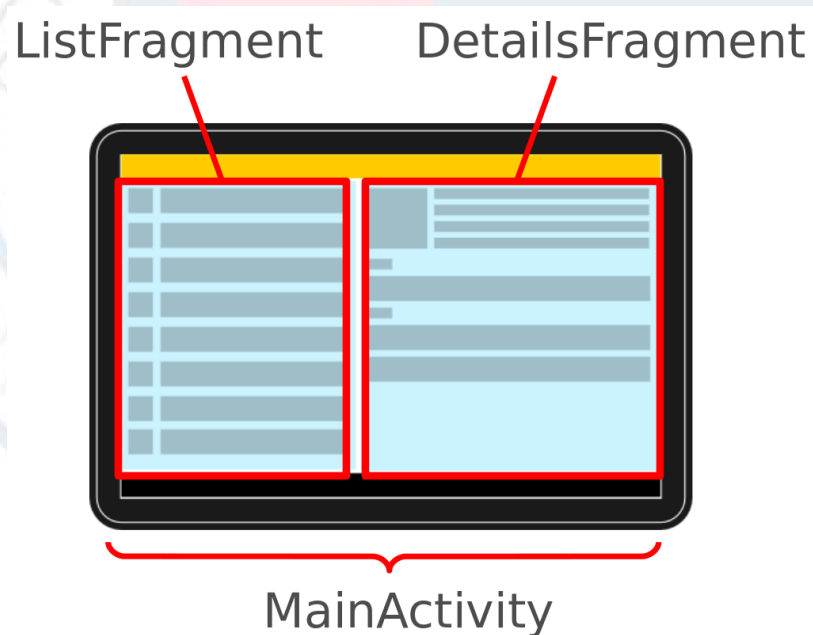
7. ការដំឡើងទំនាក់ទំនងរវាង Fragment & Fragment (ត)

ដើម្បីទទួលបាននូវ event callback មកពី fragment គឺ Host Activity ត្រូវ Implement នូវ interface ដែលបានប្រកាសនៅក្នុង Fragment Class។

```
public class MainActivity extends Activity implements
    FragmentA.OnArticleSelectedListener {
    ...
    @Override
    public void onArticleSelected(int position) {
        // Now we got position value from the FragmentA
        // Do something here !
    }
    ...
}
```

7. ការទំនាក់ទំនងរវាង Fragment & Fragment (ត)

- Example: ប្រសិនបើ News Application មួយមាន Fragment 2 នៅក្នុង Activity ដែល fragment A សម្រាប់បង្ហាញនូវ List's article និង fragment B សម្រាប់បង្ហាញនូវ article ។ បន្ទាប់មក fragment A គួរតែប្រាប់ទៅកាន់ Activity នៅពេលដែល list item ត្រូវបាន selected ដូច្នេះហើយ Activity អាចប្រាប់ fragment B ដើម្បីធ្វើការបង្ហាញនូវ article ។



8. Deliver Message to Fragment

Host Activity អាចធ្វើការបញ្ជូន messages ទៅកាន់ fragment តាមរយៈការចាប់យក Fragment Instance ដោយប្រើប្រាស់ method `findFragmentById()` បន្ទាប់មកធ្វើការហៅ Fragment's public method ដោយផ្ទាល់។

ឧទាហរណ៍៖ សន្មត់ថាមាន Activity ដែលបង្ហាញខាងលើអាចមានផ្ទុកនូវ Fragment មួយផ្សេងទៀតដែលប្រើសម្រាប់បង្ហាញ item ដែលកំណត់ដោយ data ដែលបញ្ជូនត្រលប់មកពី callback method។ ក្នុងករណីនេះ Activity អាចបញ្ជូន information ដែលទទួលបានពី callback method នោះទៅកាន់ Fragment មួយផ្សេងទៀតដែលនឹងបង្ហាញ item។

8. Deliver Message to Fragment (🔔)

```
public class MainActivity extends Activity implements
    FragmentA.OnArticleSelectedListener {
    ...
    @Override
    public void onArticleSelected(int position) {
        FragmentB artFreg = (FragmentB) getFragmentManager()
            .findFragmentById(R.id.article_fragment);
        if(artFreg != null) {    // article fragment is available
            artFreg.updateArticleView(position);
        }
    }
}
```

8. Deliver Message to Fragment (🔔)

```
else{    // we are in FragmentA and want to swap to FragmentB
    FragmentB newFrag = FragmentB.newInstance(position);
    FragmentTransaction transaction = getSupportFragmentManager()
                                    .beginTransaction();
    transaction.replace(R.id.fragment_container, newFrag);
    transaction.addToBackStack(null);
    transaction.commit();
}
}
...
}
```