

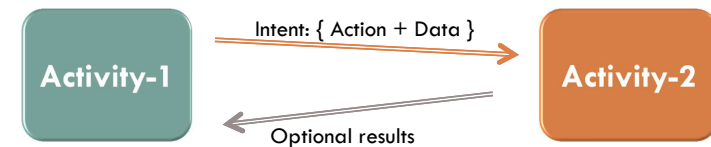
# ANDROID APPS DEVELOPMENT FOR MOBILE AND TABLET DEVICE (LEVEL I)

## Lecture 5: Intent and Fragment

Peter Lo

## Intents

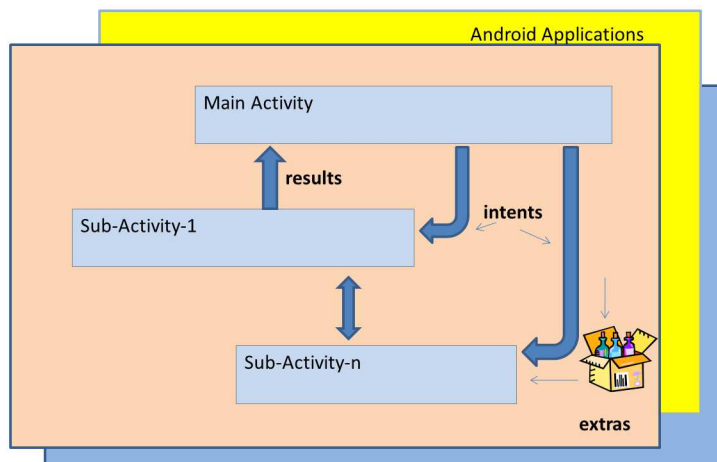
- Intents are asynchronous messages which allow application components to request functionality from other Android components.
- Intents allow you to interact with components from the own and other applications.
  - For example an activity can start an external activity for taking a picture.



X4-XT-CDP-0251-A @ Peter Lo 2015

2

## Android Applications & Activities Overview



X4-XT-CDP-0251-A @ Peter Lo 2015

3

## Creating Simple Intent

- Typically an intent is called with Action/Data pair:

```
Intent myIntent = new Intent( Intent.ACTION_VIEW,
                             Uri.parse("www.polyu.edu.hk"));
startActivity (myIntent);
```

The built-in action to be performed, such as ACTION\_VIEW, ACTION\_EDIT, ACTION\_CALL, or user-created-activity

The primary data to operate on, such as a phone number to be called (expressed as a Uri such as tel:// , http:// , sendto://)

The `startActivity()` method is used to start a new activity, which will be placed at the top of the activity stack.

X4-XT-CDP-0251-A @ Peter Lo 2015

4

## Common Intent Action/Data Pairs

Examples of **action/data** pairs are:

<b>ACTION_DIAL</b>	<b>tel:27665111</b> Display the phone dialer with the given number filled in.
<b>ACTION_SENDTO</b>	<b>smsto:27665111</b> Display the SMS editor with the given number filled in.
<b>ACTION_VIEW</b>	<b>http://www.polyu.edu.hk</b> Show specified web page in a browser view.
<b>ACTION_VIEW</b>	<b>content://contacts/people/</b> Display a list of people, which the user can browse through. Selecting a particular person to view would result in a new intent
<b>ACTION_EDIT</b>	<b>content://contacts/people/2</b> Edit information about the contact person whose identifier is "2".

## Secondary Attributes

- In addition to the primary action/data attributes, there are secondary attributes that you can also include with an intent, such as: Category, Components, Type, and Extras.

<p><b>Type</b></p> <p>Set an explicit <b>MIME</b> data type</p> <p>contacts/people images/pictures images/video audio/mp3</p> <p><small>MIME - Multipurpose Internet Mail Extensions</small></p> <p><b>Extras</b></p> <p>This is a <b>Bundle</b> of any additional information. Typical methods include: bundle.putInt(key, value) bundle.getInt(key)</p>	<p><b>Category</b></p> <p>additional information about the action to execute</p> <ul style="list-style-type: none"> <li>☑ CATEGORY_ALTERNATIVE: String - Intent</li> <li>☑ CATEGORY_APP_BROWSER: String - Intent</li> <li>☑ CATEGORY_APP_CALCULATOR: String - Intent</li> <li>☑ CATEGORY_APP_CALENDAR: String - Intent</li> <li>☑ CATEGORY_APP_CONTACTS: String - Intent</li> <li>☑ CATEGORY_APP_EMAIL: String - Intent</li> <li>☑ CATEGORY_APP_GALLERY: String - Intent</li> <li>☑ CATEGORY_APP_MAPS: String - Intent</li> <li>☑ CATEGORY_APP_MARKET: String - Intent</li> <li>☑ CATEGORY_APP_MESSAGING: String - Intent</li> <li>☑ CATEGORY_APP_MUSIC: String - Intent</li> </ul> <p><b>Component</b></p> <p>Explicit name of a component class to use for the intent (eg. "MyMethod1")</p>
---	---

X4-XT-CDP-0251-A @ Peter Lo 2015

6

## Creating Intent with Secondary Attribute

- We can pass the secondary attribute with the **putExtra()** method:

```
Intent myIntent = new Intent( Intent.ACTION_WEB_SEARCH );
myIntent.putExtra( SearchManager.QUERY, "PolyU" );
startActivity(myIntent);
```

The built-in action to be performed, such as ACTION\_VIEW, ACTION\_EDIT, ACTION\_CALL, or user-created-activity

The secondary data to operate on, such as passing a string as an Extra argument for a Google Search. The string is a 'human' query with keywords.

The **startActivity()** method is used to start a new activity, which will be placed at the top of the activity stack.

X4-XT-CDP-0251-A @ Peter Lo 2015

7

## Starting Activities and Getting Results

- In order to get results back from the called activity we use the method

**startActivityForResult ( Intent, requestCodeID )** →

- Where requestCodeID is an arbitrary value you choose to identify the call (similar to a 'nickname').
- The result sent by the sub-activity could be picked up through the listener-like asynchronous method

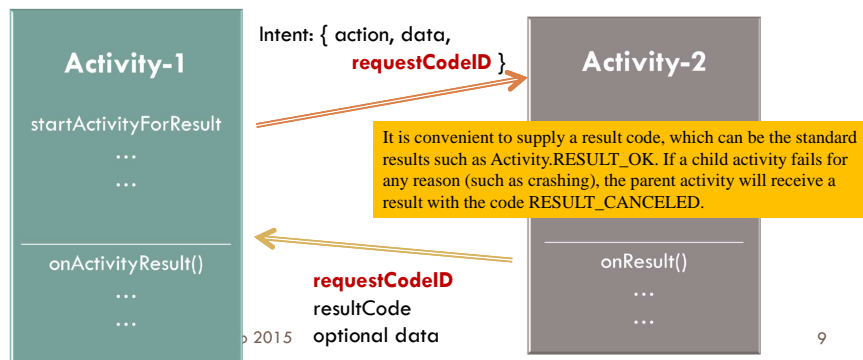
**onActivityResult ( requestCodeID, resultCode, Intent )** ←

X4-XT-CDP-0251-A @ Peter Lo 2015

8

## Starting Activities and Getting Results

- Before an invoked activity exits, it can call `setResult()` to return a termination signal back to its parent.
- All of this information can be capture back on the parent's `onActivityResult()`.



9

## Android Bundles

- Android applications may include multiple inter-active activities.
- The Android Bundle container is a simple mechanism used to pass data between co-operating activities.
- A Bundle is a type-safe MAP collection of <name, value> pairs.
- There is a set of `put` and `get` methods to store and retrieve (single and array) values of primitive data types from/to the bundles.

```
Bundle myBundle = new Bundle();
myBundle.putDouble ("var1", 3.1415);
...
Double v1 = myBundle.getDouble("var1");
```

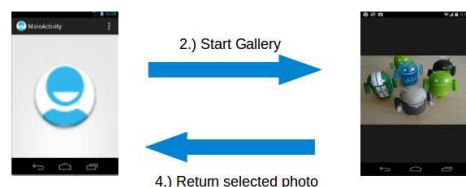
X4-XT-CDP-0251-A @ Peter Lo 2015

10

## Fragments

- Android introduced fragments in Android 3.0, primarily to support more dynamic and flexible UI designs on large screens, such as tablets.
- By dividing the layout of an activity into fragments, you become able to modify the activity's appearance at runtime and preserve those changes in a back stack that's managed by the activity.

You can think of a fragment as a modular section of an activity, which has its own lifecycle, receives its own input events, and which you can add or remove while the activity is running



X4-XT-CDP-0251-A @ Peter Lo 2015

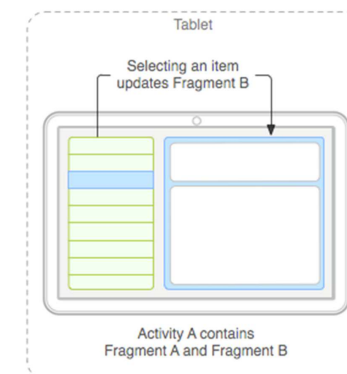
1.) User triggers "Pick photo" via button

3.) User selects photo

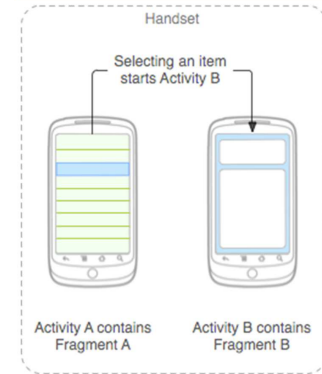
11

## Approach for Creating Fragment

### Use One Activities



### Use Separate Activities

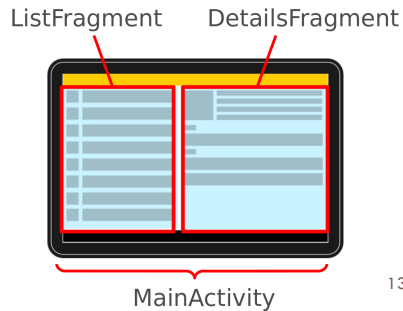


X4-XT-CDP-0251-A @ Peter Lo 2015

12

## Create Fragment by One Activities

- Use one activity, which displays two fragments for tablets and on handset devices.
- You would switch the fragments in the activity whenever necessary. This requires that the fragment is not declared in the layout file as such fragments cannot be removed during runtime.

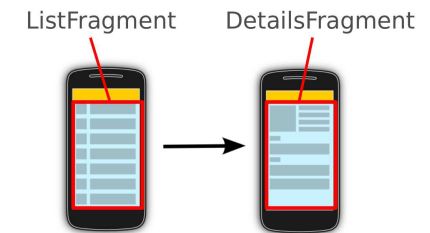


X4-XT-CDP-0251-A @ Peter Lo 2015

13

## Create Fragment by Separate Activities

- Use separate activities to host each fragment on a handset.
- When the tablet UI uses two fragments in an activity, use the same activity for handsets, but supply an alternative layout that includes just one fragment.
- When you need to switch fragments, start another activity that hosts the other fragment.



X4-XT-CDP-0251-A @ Peter Lo 2015

14

## Create a Fragment Class

- To create a fragment, extend the Fragment class, then override key lifecycle methods to insert your app logic, similar to the way you would with an Activity class.
- You must use the `onCreateView()` callback to define the layout. In fact, this is the only callback you need in order to get a fragment running.

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

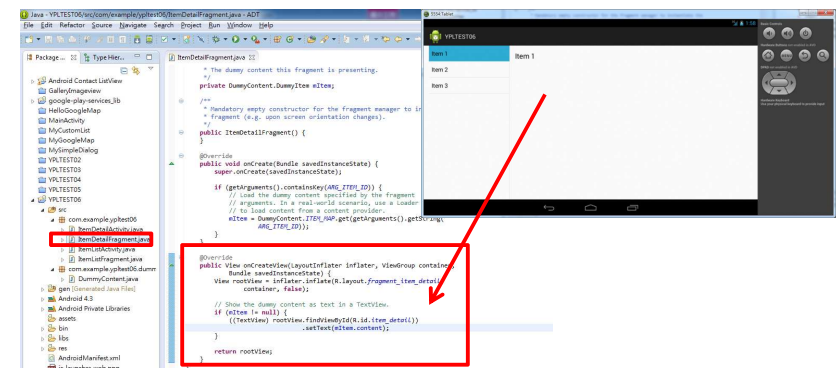
public class ArticleFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.article_view, container, false);
    }
}
```

X4-XT-CDP-0251-A @ Peter Lo 2015

15

## Master/Detail Fragment Template

- If using the default template, you can customize the code in "ItemDetailFragment.java"



X4-XT-CDP-0251-A @ Peter Lo 2015

16