Fragments


Repo: https://github.com/karlmorris/AndroidFragments/tree/Communication
Overview

• Communication among Fragments and Activities
Fragments

- Allow for the creation of dynamic UI
  - Responsive to different devices and screen orientations
- Must practice proper OO principles of low coupling, high cohesion
Fragments
Fragments

• Limit dependency among fragments
• Communicate only through parent activity
  – Fragment $\rightarrow$ Activity: interface
  – Activity $\rightarrow$ Fragment: public methods
Fragment Communication

• You may require various communication scenarios in an application
  – Fragment to Activity
  – Activity to Fragment
  – Fragment to Fragment

• Proper fragment design dictates that all communication must pass through activity
  – Fragments should only know about their parent activity through an interface
Fragment Communication

- Communication scenarios become:
  - Fragment to activity
  - Activity to Fragment
  - Fragment A to Activity to Fragment B

- This respects high cohesion and low coupling
Fragment to Activity

- Communication from a fragment to a parent activity is facilitated by interfaces
- The fragment defines an interface which is implemented by the activity
- Various methods are then defined for each type of interaction or message. The activity implements these methods to receive the message/data
Activity to Fragment

- Communication from an activity to a fragment is facilitated by **public methods**
- These public methods serve as an API that the activity can invoke on the fragment instance
- Method calls are similar to those on other objects
Fragment to Activity to Fragment

- This communication is facilitated by a combination of the **Fragment to Activity** and the **Activity to Fragment** patterns.

- This pattern is required because no fragment instance should have a reference to another fragment instance:
  - Improper design principles
  - Potential memory leaks