Android Workshop

MIT COE FOSS GROUP

Gaurav Deshmukh

Devaj Mitra

Broadcast Receivers

- Receive and react to broadcast announcements
- Extend the class BroadcastReceiver
- Examples of broadcasts:
- Low battery, power connected, shutdown, timezone changed, etc.
- Other applications can initiate broadcasts

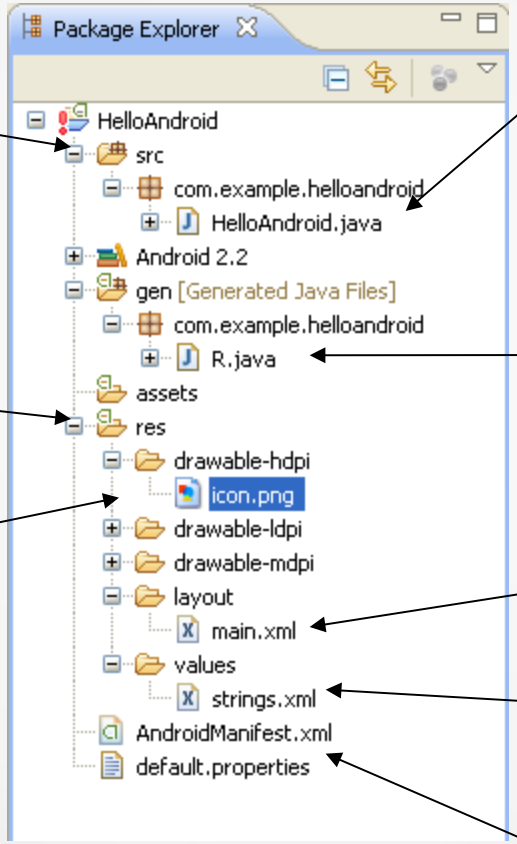
Content Provider

- Makes some of the application data available to other applications
- It’s the only way to transfer data between applications in Android (no shared files, shared memory, pipes, etc.)
- Extends the class ContentProvider;
- Other applications use a ContentResolver object to access the data provided via a ContentProvider

Hello World

All source code here

Java code for our activity



All non-code resources

Generated Java code
Helps link resources to Java code

Images

Layout of the activity

Strings used in the program

Android Manifest

XML

- Used to define some of the resources
  - Layouts (UI)
  - Strings
- Manifest file
- Shouldn’t usually have to edit it directly, Eclipse can do that for you
- Preferred way of creating UIs
  - Separates the description of the layout from any actual code that controls it
  - Can easily take a UI from one platform to another

R Class

- Auto-generated: you shouldn’t edit it
- Contains IDs of the project resources
- Enforces good software engineering
- Use findViewById and Resources object to get access to the resources
  - Ex. Button b = (Button)findViewById(R.id.button1)
  - Ex. getResources().getString(R.string.*hello*));

Layouts

- Eclipse has a great UI creator
  - Generates the XML for you
- Composed of *View* objects
- Can be specified for portrait and landscape mode
  - Use same file name, so can make completely different UIs for the orientations without modifying any code

Strings

- In res/values
  - strings.xml
- Application wide available strings
- Promotes good software engineering
- UI components made in the UI editor should have text defined in strings.xml

- Strings are just one kind of ‘Value’ there are many others

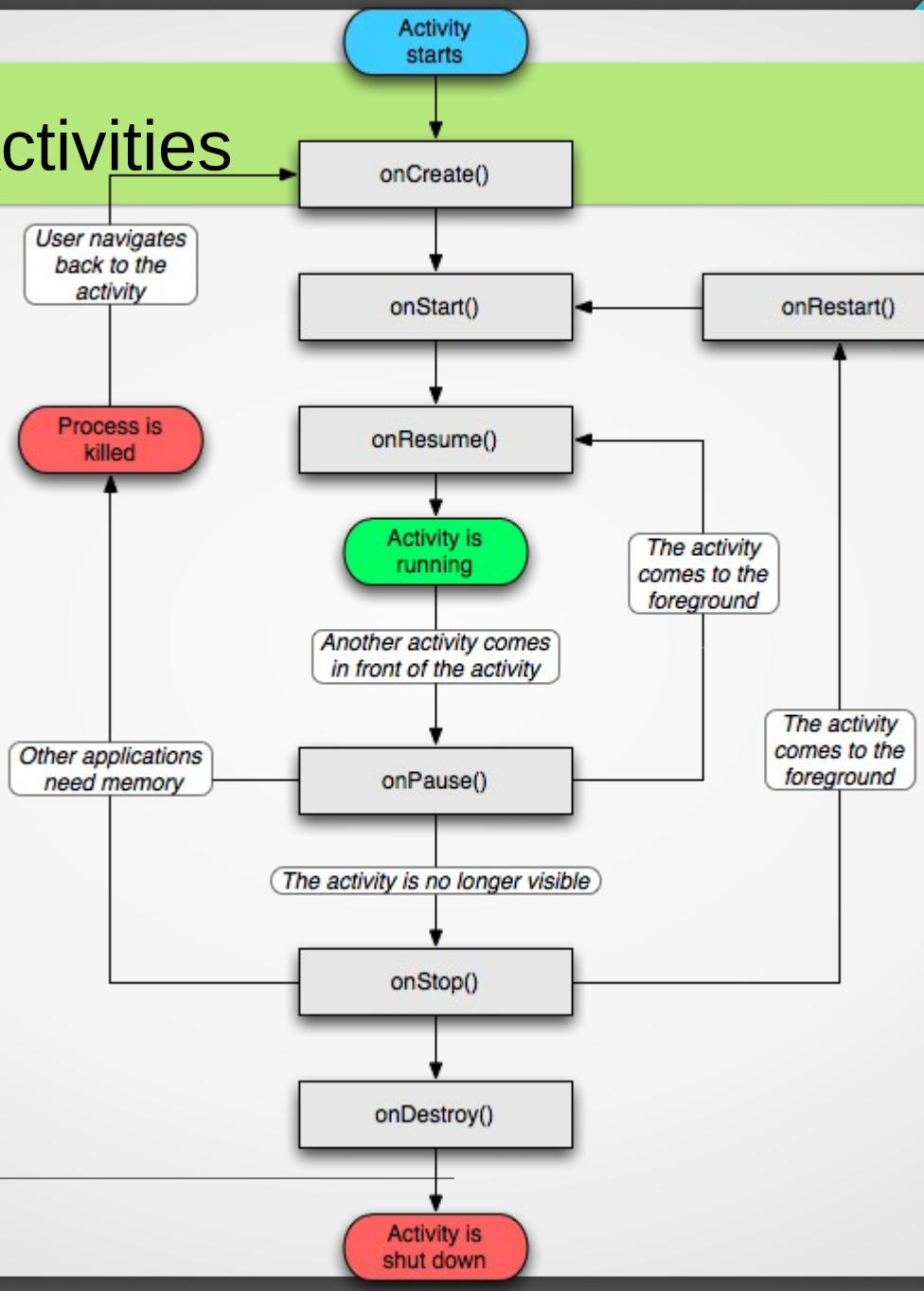
Manifest File

- Contains characteristics about your application
- When have more than one Activity in app, NEED to specify it in manifest file
  - Go to graphical view of the manifest file
  - Add an Activity in the bottom right
  - Browse for the name of the activity
- Need to specify Services and other components too
- Also important to define permissions and external libraries, like Google Maps API

Activities

- The basis of android applications
- A single Activity defines a single viewable screen
  - the actions, not the layout
- Can have multiple per application
- Each is a separate entity
- They have a structured life cycle
  - Different events in their life happen either via the user touching buttons or programmatically

Life Cylce Activities



Intents

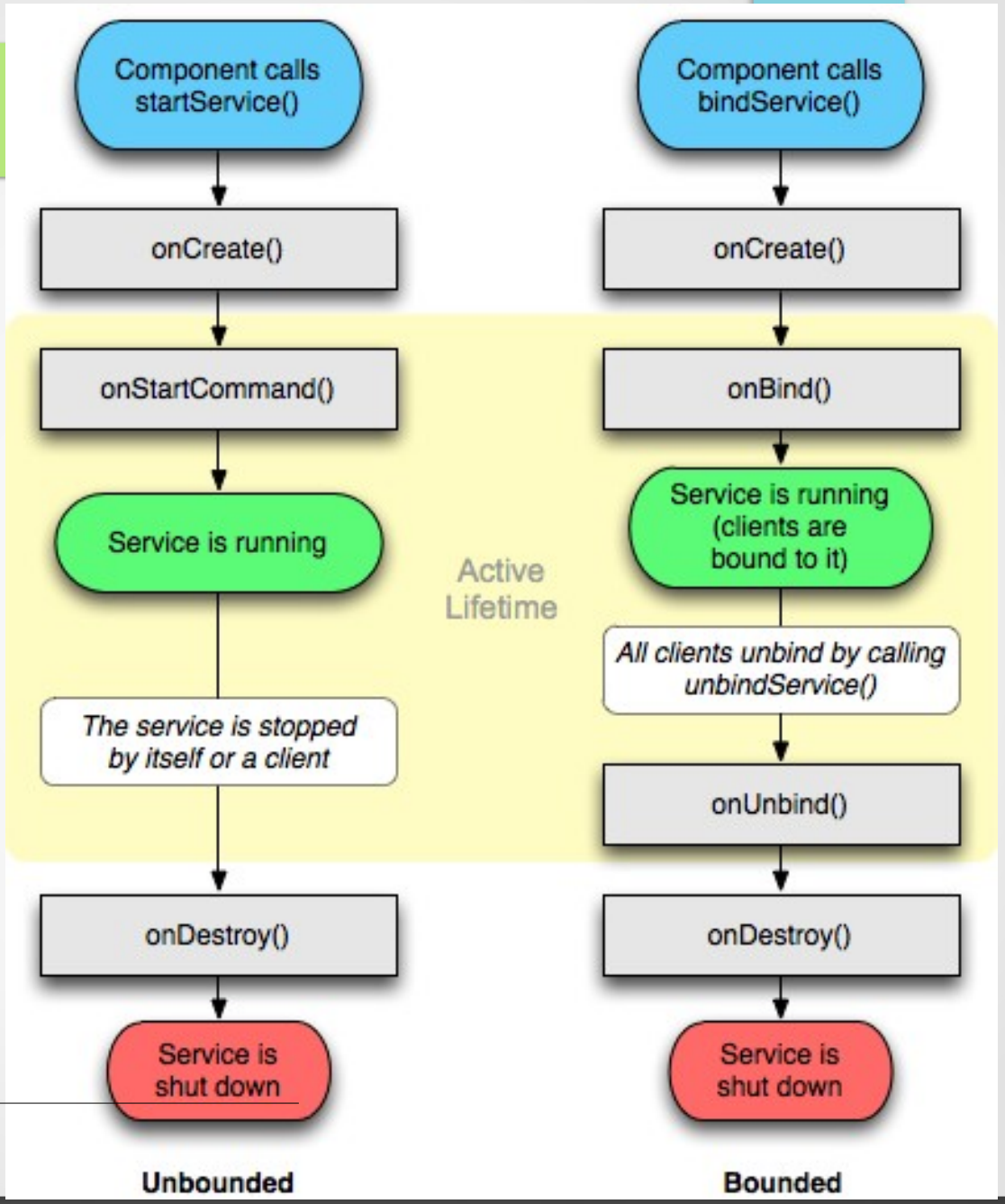
- An intent is an Intent object with a message content.
- Activities, services and broadcast receivers are started by intents. ContentProviders are started by ContentResolvers:

- An activity is started by Context.startActivity(Intent intent) or Activity.startActivityForResult(Intent intent, int RequestCode)
- A service is started by Context.startService(Intent service)
- An application can initiate a broadcast by using an Intent in any of Context.sendBroadcast(Intent intent), Context.sendOrderedBroadcast(), and Context.sendStickyBroadcast()

Services

- Run in the background
  - Can continue even if Activity that started it dies
  - Should be used if something needs to be done while the user is not interacting with application
    - Otherwise, a thread is probably more applicable
  - Should create a new thread in the service to do work in, since the service runs in the main thread
- Can be bound to an application
  - In which case will terminate when all applications bound to it unbind
  - Allows multiple applications to communicate with it via a common interface
- Needs to be declared in manifest file
- Like Activities, has a structured life cycle

Services


Component calls startService()
onCreate()
onStartCommand()
Service is running
The service is stopped by itself or a client
onDestroy()
Service is shut down
Unbounded
Component calls bindService()
onCreate()
onBind()
Service is running (clients are bound to it)
All clients unbind by calling unbindService()
onUnbind()
onDestroy()
Service is shut down
Bounded
Active Lifetime

Let's Code!!

Debugging

- Instead of using traditional System.out.println, use the Log class
  - Imported with android.util.Log
  - Multiple types of output (debug, warning, error, …)
  - Log.d(<tag>,<string>)
- Can be read using logcat.
  - Print out the whole log, which auto-updates
    - adb logcat
  - Erase log
    - adb logcat –c
  - Filter output via tags
    - adb logcat <tag>:<msg type> *:S
    - can have multiple <tag>:<msg type> filters
    - <msg type> corresponds to debug, warning, error, etc.
    - If use Log.d(), then <msg type> = D
- Reference
  - http://developer.android.com/guide/developing/debugging/debugging-log.html

About Us

Devaj Mitra

Gaurav Deshmukh

- Co-Founders @ Inorex Studio

www.inorexstudio.com

For more Android tutorials follow us on

www.facebook.com/inorexstudio