

# Základní pojmy pro vývoj aplikací na Android

Michal Jirků



# Obsah

- Architektura
- App Components
- Activities
- Services
- Content providers
- Broadcast receivers
- Intents
- Resources



# Android Anatomy



## APPLICATIONS



## APPLICATION FRAMEWORK



## LIBRARIES



## ANDROID RUNTIME



## LINUX KERNEL



# App Components

- Základní stavební bloky aplikací
- Každý komponent specifickou úlohu
- Úzce spjato s `AndroidManifest.xml`
- Čtyři základní komponenty



# Activities

- Aktuální obrazovka se kterou uživatel interaguje (např. seznam přijatých zpráv)
- Každá aplikace má většinou více aktivit (přijaté, odeslané a napsat zprávu atd...)
- V jeden okamžik běží jen jedna aktivita
- Při spuštění nové aktivity, stará přejde do „back stacku“, FIFO



# Activities

- Existence ve třech stavech  
Resumed  
Paused  
Stopped
- O správu životního cyklu se stará  
ActivityManager



# Activities

```
public class MainActivity extends Activity {
    String msg = "Android : ";

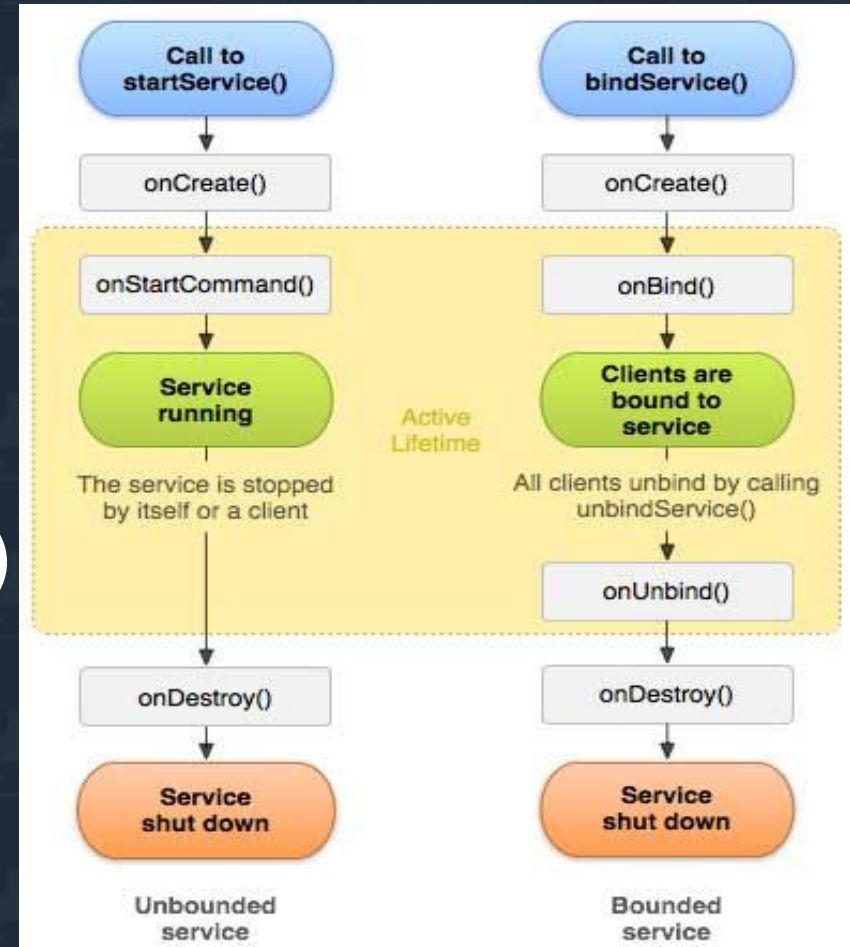
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(msg, "The onCreate() event");
    }

    /** Called when the activity is about to become visible. */
    @Override
    protected void onStart() {
        super.onStart();
        Log.d(msg, "The onStart() event");
    }

    /** Called when the activity has become visible. */
    @Override
    protected void onResume() {
        super.onResume();
        Log.d(msg, "The onResume() event");
    }
}
```

# Services

- Běží na pozadí
- Dlouhotrvající úkoly, přístup ke vzdáleným zdrojům atd.  
(Např. hudba na pozadí)
- Dva stavy:  
Started X Bound





# Services

```
public class HelloService extends Service {

    /** indicates how to behave if the service is killed */
    int mStartMode;
    /** interface for clients that bind */
    IBinder mBinder;
    /** indicates whether onRebind should be used */
    boolean mAllowRebind;

    /** Called when the service is being created. */
    @Override
    public void onCreate() {

    }

    /** The service is starting, due to a call to startService() */
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        return mStartMode;
    }

    /** A client is binding to the service with bindService() */
    @Override
    public IBinder onBind(Intent intent) {
        return mBinder;
    }
}
```

# Content providers

- Na požádání poskytuje data ( Např. telefonní seznam)
- Avšak aplikace musí mít řádné povolení
- Data jsou různými způsoby ukládána (databáze, soubory, přes síť)



# Broadcast receivers

- Reaguje na vysílané zprávy od ostatních aplikací nebo systému
- Aplikace mohou vysílat broadcasty, ostatní aplikace tak dostanou na vědomí, že se např. soubor stáhnul a je tak pro ně k dispozici



# Broadcast receivers

- Vytvoření a registrace

```
public class MyReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Toast.makeText(context, "Intent Detected.", Toast.LENGTH_LONG).show();  
    }  
  
}
```

```
<application  
    android:icon="@drawable/ic_launcher"  
    android:label="@string/app_name"  
    android:theme="@style/AppTheme" >  
  
    <receiver android:name="MyReceiver">  
        <intent-filter>  
            <action android:name="android.intent.action.BOOT_COMPLETED">  
        </action>  
        </intent-filter>  
    </receiver>  
  
</application>
```



# Intents

- Intent je objekt nesoucí nějakou žádost
- Posílá se z jedné komponenty k druhé
- Explicitní X Implicitní

```
// Explicit Intent by specifying its class name
Intent i = new Intent(this, TargetActivity.class);
i.putExtra("Key1", "ABC");
i.putExtra("Key2", "123");

// Starts TargetActivity
startActivity(i);
```

```
// Implicit Intent by specifying a URI
Intent i = new Intent(Intent.ACTION_VIEW,
Uri.parse("http://www.example.com"));

// Starts Implicit Activity
startActivity(i);
```

# Resources

- Externí elementy: obrázky, zvuky, text,...
- SDK přiřazuje ID k použití jako referenci
- logo.png (v res/drawable/ ), pak SDK vygeneruje resource ID R.drawable.logo
- Ulehčení práce s kódováním: aplikace je ve více jazycích, kompatibilita zobrazení atd...



# Resources

Příklad:

- Mějme res/values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello, World!</string>
</resources>
```

- Pak můžeme nastavit text na TextView objektu pomocí resource ID

```
TextView msgTextView = (TextView) findViewById(R.id.msg);
msgTextView.setText(R.string.hello);
```



OTÁZKY?





# Zdroje

- <http://i42.tinypic.com/2r4u8zt.png>
- <http://developer.android.com/guide/components/fundamentals.html>
- <http://www.tutorialspoint.com/android/index.htm>

