Android Architecture and Development

Éric Languénou

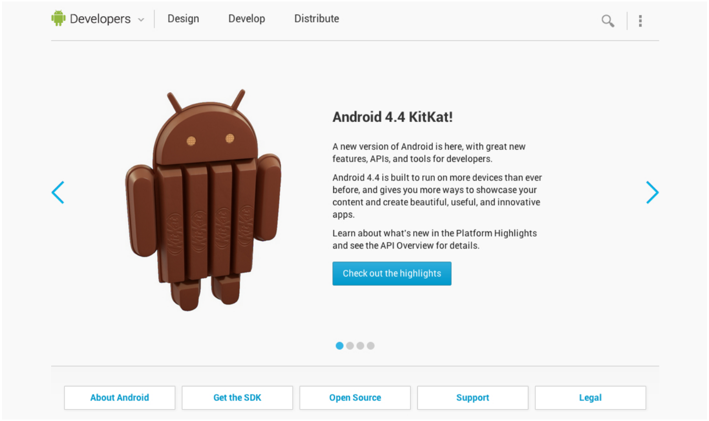
2013-2014

UNIVERSITÉ DE NANTES

The Big Picture

- What is Android?
- Overview development environment
- ressources : `http://www.android.com/`

Android Developper



Android 4.4 KitKat!

A new version of Android is here, with great new features, APIs, and tools for developers.

Android 4.4 is built to run on more devices than ever before, and gives you more ways to showcase your content and create beautiful, useful, and innovative apps.

Learn about what's new in the Platform Highlights and see the API Overview for details.

What is Android?

- Android is an open-source software platform created by Google and the Open Handset Alliance.
- It is primarily used to power mobile phones.
- It has the capability to make inroads in many other (non-phone) embedded application markets.

What is Android? (2)

- AndroidTM consists of a complete set of software components for mobile devices including:
  - an operating system, - middleware, and embedded key mobile applications
  - a large market.

Why Android?

- often seen as alternative to iPhone
- based on linux
- openSource (apache) and free

The Android Platform

- Android is a **software environment** built for mobile devices.
- It is not a hardware platform.
- Android includes:
  - Linux kernel-basedOS,
  - a rich UI,
  - telephone functionality,
  - end-user applications,
  - code libraries,
  - application frameworks
  - multimedia support, ...
- User applications are built for Android in Java.

Android's Context: Mobile Market Player

- Stakeholders:
- Mobile**network operators** want Operators to lock down their networks, controlling and metering traffic.
- **Device manufacturers** want to differentiate themselves with features, reliability, and price points.
- **Software vendors** want complete access to the hardware to deliver cutting-edge applications.

The Maturing Mobile Experience

- Not so long ago ..
  - phone
  - pagers
  - PDA organizer
  - Laptop
  - portable music player
  - no internet access
- Today
  - smartphone
  - tablets
  - pha-blets
- Tomorrow ?

The Maturing Mobile Experience

UNIVERSITÉ DE NANTES

I want my 2015 Smartphone to act as ..

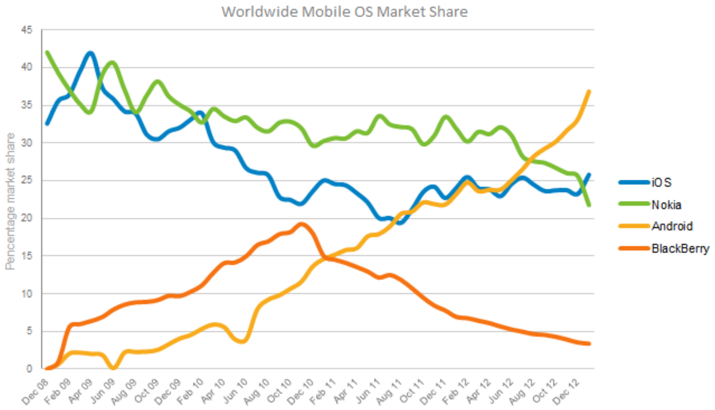
- Phone
- Pager
- PDA Organizer
- High Quality Camera (still & video)
- Portable music player
- Portable TV / Video Player / Radio
- Laptop
- Play Station
- GPS
- Golf Caddy (ball retriever too)
- Book Reader (I don't read, It reads to me)
- Car / Home / Office Key
- Remote Control (Garage, TV, ...)
- Credit Card / Driver's License / Passport
- Cash on Demand 16. Cook, house chores
- Psychologist / Mentor / Adviser
- ????

Android vs. Competitors

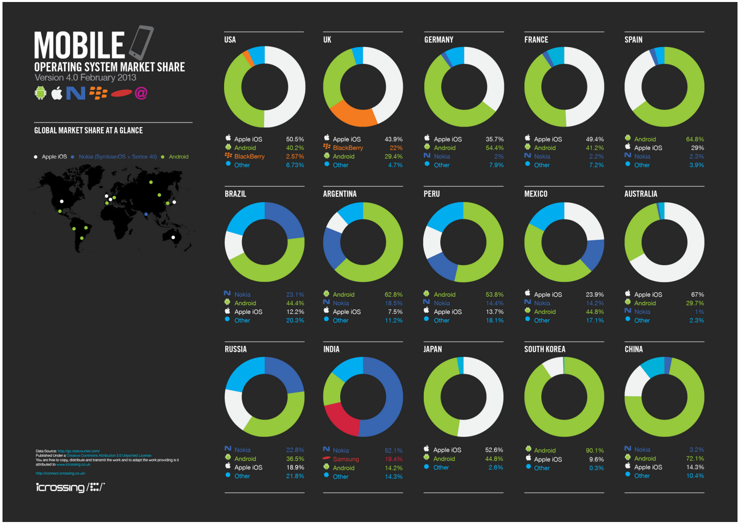
UNIVERSITÉ DE NANTES

- Apple Inc.
- Microsoft
- Nokia
- Palm
- BlackBerry
- Research In Motion
- Symbian

Mobile Market



Mobile Market(2)



USA

Apple iOS	50.5%
Android	40.2%
BlackBerry	2.57%
Other	6.73%

UK

Apple iOS	43.9%
BlackBerry	22%
Android	29.4%
Other	4.7%

GERMANY

Apple iOS	35.7%
Android	54.4%
Nokia	2%
Other	7.9%

FRANCE

Apple iOS	49.4%
Android	41.2%
Nokia	2.2%
Other	7.2%

SPAIN

Android	64.8%
Apple iOS	29%
Nokia	2.3%
Other	3.9%

BRAZIL

Nokia	23.1%
Android	44.4%
Apple iOS	12.2%
Other	20.3%

ARGENTINA

Android	62.8%
Nokia	18.5%
Apple iOS	7.5%
Other	11.2%

PERU

Android	53.8%
Nokia	14.4%
Apple iOS	13.7%
Other	18.1%

MEXICO

Apple iOS	23.9%
Nokia	14.2%
Android	44.8%
Other	17.1%

AUSTRALIA

Apple iOS	67%
Android	29.7%
Nokia	1%
Other	2.3%

RUSSIA

Nokia	22.8%
Android	36.5%
Apple iOS	18.9%
Other	21.8%

INDIA

Nokia	52.1%
Samsung	19.4%
Android	14.2%
Other	14.3%

JAPAN

Apple iOS	52.6%
Android	44.8%
Other	2.6%

SOUTH KOREA

Android	90.1%
Apple iOS	9.6%
Other	0.3%

CHINA

Nokia	3.2%
Android	72.1%
Apple iOS	14.3%
Other	10.4%

Android Components (Stack)

- The Android stack includes a large array of features for mobile applications.
- It would be easy to confuse Android with a general purpose computing environment.
- All of the major components of a computing platform are included.

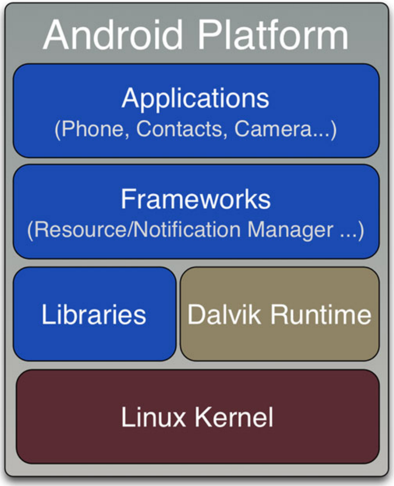
Android Components

UNIVERSITÉ DE NANTES

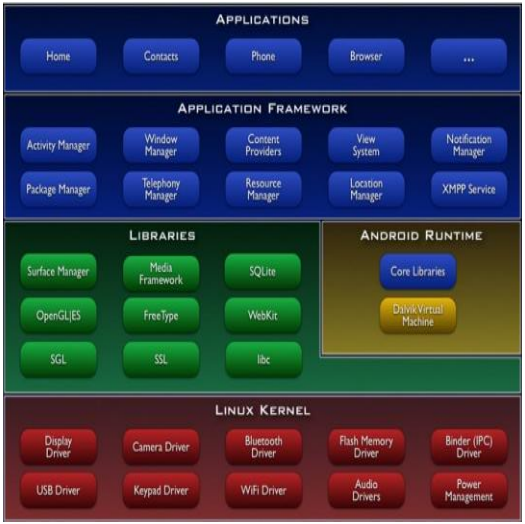
- Application framework enabling reuse and replacement of components
- Dalvik virtual machine optimized for mobile devices
- Integrated browser based on the open source WebKit engine
- Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES specification (hardware acceleration optional)
- SQLite for structured data storage
- Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM Telephony (hardware dependent)
- Bluetooth, EDGE, 3G, 4G, and Wi-Fi (hardware dependent)
- Camera, GPS, compass, and accelerometer (hardware dependent)
- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

Android Components

UNIVERSITÉ DE NANTES



Android Components(2)



Why use Linux for a phone?

UNIVERSITÉ DE NANTES

- Linux kernel is a proven core platform. comes to a mobile phone, because voice communication is the primary use of a phone.
- Linux provides a hardware abstraction layer, letting the upper levels remain unchanged despite changes in the underlying hardware.
- As new accessories appear on the market, drivers can be written at the Linux level to provide support, just as on other Linux platforms.

Dalvik Virtual Machine

- User applications, as well as core Android applications, are written in Java programming language and are compiled into byte codes.
- Android byte codes are interpreted at runtime by a processor known as the Dalvik virtual machine.

Why another JavaVirtual Machine?

- Android bytecode files are logically equivalent to Java bytecodes, but they permit Android to :
  - run its applications in its own virtual environment that is free from Sun's licensing restrictions and
  - an open platform upon which Google, and potentially the open source community, can improve as necessary.

Inside Android: Intents

- An important and recurring theme of Android
- An Intent in Android describes what you want to do.
- This may look like :
  - "I want to look up a contact record," or
  - "Please launch this website," or
  - "Show the Order Confirmation Screen."
- Intents are important because they facilitate navigation and represent the most important aspect of Android coding.

Intents and IntentFilters

- An Intent is a declaration of need.
- An Intent is made up of various pieces including:
  - desired action or service,
  - data, and
  - category of component that should handle the intent and instructions on how to launch a target activity.
- An IntentFilter is a trigger, a declaration of capability and interest in offering assistance to those in need.
- An IntentFilter may be generic or specific with respect to which Intents it offers to service.

Intents and IntentFilters (2)

- An intent is an abstract description of an operation to be performed.
- Its most significant use is in the launching of activities, where it can be thought of as the glue between activities.
- The primary pieces of information in an intent are:
  - Action : like `ACTION_VIEW,ACTION_EDIT,ACTION_MAIN`
  - Data : the necessary data to operate: database, contacts, expressed as url

Intents and IntentFilters (3)

Some examples of Intent's action/data pairs are:

- `ACTION_VIEW` content://contacts/1 – Display information about the person whose identifier is "1".
- `ACTION_DIAL` content://contacts/1 – Display the phone dialer with the person filled in.
- `ACTION_VIEW` tel:123 – Display the phone dialer with the given number filled in
- `ACTION_DIAL` tel:123 – Display the phone dialer with the given number filled in.
- `ACTION_EDIT` content://contacts/1 – Edit information about the person whose identifier is "1".
- `ACTION_VIEW` content://contacts/ – Display a list of people, which the user can browse through.

Dissecting Intents

1. Component name The name of the component that should handle the intent (for example "com.example.project.app.MyActivity1").
2. Category A string containing additional information about the kind of component that
3. Action A string naming the action to be performed ? or, in the case of broadcast intents, the action that took place and is being reported (for example: `ACTION_VIEW`, `ACTION_CALL`, `ACTION_TIMEZONE_CHANGED`, `...`).
4. Data The URI of the data to be acted on and the MIME type of that data (for example tel:/216 555-1234 , "http://maps.google.com", ...). should handle the intent (for example `CATEGORY_BROWSABLE`, `CATEGORY_LAUNCHER`, `...`).
5. Extras Key-value pairs for additional information that should be delivered to the component handling the intent.
6. Flags of various sorts.

Delivering Intents

- An Intent object is passed to to launch an activity or get an existing activity to do something new (asynchronous & synchronously respectively).
- An Intent object is passed to Context.startService() to initiate a service or deliver new instructions to an ongoing service.
- An intent can be passed to Context.bindService() to establish a connection between the calling component and a target service. It can optionally initiate the service if it's not already running.

Intent Resolution

Intents can be divided into two groups:

- Explicit intents designate the target component by its name, typically used for an activity starting a subordinate service or launching a sister activity.
- Implicit intents do not name a target (the field for the component name is blank). Implicit intents are often used to activate components in other applications. Late binding applies.

Whenever possible Android delivers an explicit intent to an instance of the designated target class.

Example of Intent

Following fragments calls an Intent whose job is to invoke a built-in task (`ACTION_VIEW`) and explore the Contacts available in the phone.

```
Intent myIntent = new Intent(
 Intent.ACTION\_VIEW,
 Uri.parse("content://contacts/people"));
startActivity(myIntent);
```

Example of Intent, more

Complete code to view contacts:

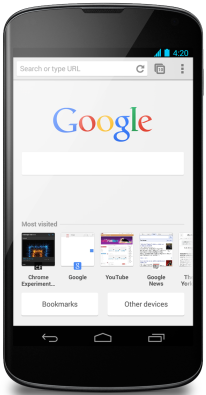
```java
package matos.cis493;
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
public class AndDemo1 extends Activity {
/** show contact list */ @Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
Intent myIntent = new Intent( Intent.ACTION_VIEW,Uri.parse
    ( "content://contacts/people"));
startActivity(myIntent);
}
}
```

Other Example of Intent

UNIVERSITÉ DE NANTES

- Following Intent uses built-in task(`ACTION_VIEW`) to explore a web page (see new Uri value)

```
Intent myIntent = new Intent( Intent.ACTION_VIEW, Uri
     .parse("http://www.google.com"));
startActivity(myIntent);
```

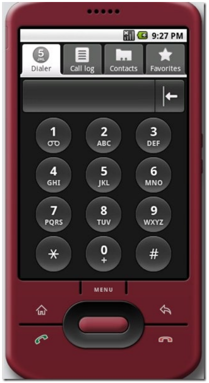


Other Example of Intent

UNIVERSITÉ DE NANTES

- Following Intent uses built-int ask(`ACTION_VIEW`) to make a phone call (see new Uri value)

```
Intent myIntent = new Intent( Intent.ACTION\_VIEW,
    Uri.parse("tel:/216 555-1234"));
startActivity(myIntent);
```



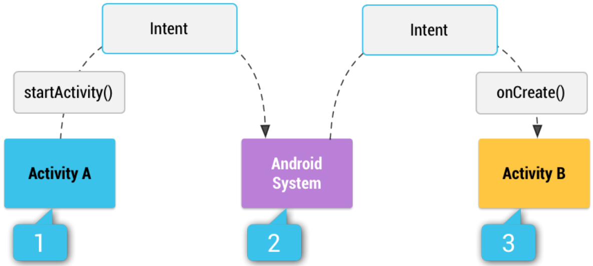
IntentFilters

- The **IntentFilter** defines the relationship between the Intent and the application.
- **IntentFilters** can be specific to the data portion of the Intent, the action portion, or both.
- **IntentFilters** also contain a field known as a category. A category helps classify the action.
- For example, the category named `CATEGORY_LAUNCHER` instructs Android that the Activity containing this IntentFilter should be visible in the home screen.

IntentFilters

- When an Intent is dispatched, the system evaluates the available Activities, Services, and registered BroadcastReceivers and routes the Intent to the most appropriate recipient

IntentFilters



- Illustration of how an **implicit intent** is delivered through the system to start another activity
  - Activity A creates an Intent with an action description and passes it to startActivity().
  - The Android System searches all apps for an intent filter that matches the intent. When a match is found,
  - the system starts the matching activity (Activity B) by invoking its onCreate() method and passing it the Intent.

IntentFilters

- To inform the system which **implicit intents** they can handle, *activities, services, and broadcast receivers* can have one or more intent filters.
- Each filter describes a capability that the component is willing to receive.
- An **explicit intent** is always delivered to its target, no matter what it contains; the filter is not consulted.
- But an implicit intent is delivered to a component only if it can pass through one of the component's filters.

IntentFilters

- For example, here's an activity declaration with an intent filter to receive an `ACTION_SEND` intent when the data type is text:

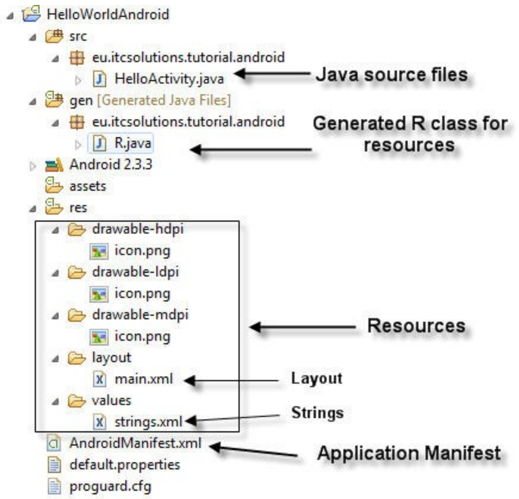
```
<activity android:name="ShareActivity">
    <intent-filter>
        <action android:name="android.intent.action.
            SEND"/>
        <category android:name="android.intent.
            category.DEFAULT"/>
        <data android:mimeType="text/plain"/>
    </intent-filter>
</activity>
```

- It's okay to create a filter that includes more than one instance of `<action>, <data>, or <category>`.
- If you do, you simply need to be certain that the component can handle any and all combinations of those filter elements.

Android Applications

- Each Android application runs in its own Linux process.
- An application consists of a combination of software components including:
  - Activities
  - Services
  - Broadcast Receivers
  - Content Providers

Structure of a Typical Android App



```
HelloWorldAndroid
  src
    eu.itcsolutions.tutorial.android
      HelloActivity.java
  gen [Generated Java Files]
    eu.itcsolutions.tutorial.android
      R.java
  Android 2.3.3
  assets
  res
    drawable-hdpi
      icon.png
    drawable-ldpi
      icon.png
    drawable-mdpi
      icon.png
    layout
      main.xml
    values
      strings.xml
  AndroidManifest.xml
  default.properties
  proguard.cfg
```

Android Services

- A **Service** is an application component that runs for an indefinite period of time.
- Each service class must have a corresponding `<service>` declaration in its package's `AndroidManifest.xml`.
- Services can be started/stopped with
  - `Context.startService()` and
  - `Context.bindService()`.
  - `stopService(...)` and `unbindService(...)`

Android Services

- Services, like other application objects, run in the main thread of their hosting process.
- This means that :
  - if your service is going to do any CPU intensive (such as MP3 playback) or blocking (such as networking, RSS exchange) operations,
  - it should spawn its own thread in which to do that work

Android Broadcast Receiver

- What is a `BROADCASTRECEIVER`?
  - If an application wants to receive and respond to a global event (such as the phone ringing or an incoming text message) it must register as a BroadcastReceiver.
  - An application registers to receive Intents by announcing in the `AndroidManfest.xml` file its IntentFilters.
  - If the receiver is registered in the `AndroidManifest.xml` file, it does not have to be running in order to be triggered. When the global event occurs, the application is started automatically upon notification of the triggering event. All of this housekeeping is managed by the Android OS itself.
  - An application may register at runtime via the Context class's registerReceiver method.

Android Broadcast Receiver

- `BROADCASTRECEIVER` and UI.
  - Of even more importance, the code running in the onReceive method of a BroadcastReceiver should make no assumptions about persistence or long-running operations.
  - If the BroadcastReceiver requires more than a trivial amount of code execution, it is recommended that the code initiate a request to a Service to complete the requested functionality.

Intents vs. Broadcasts

- Starting an Activity with an Intent is a foreground operation that modifies what the user is currently interacting with.
- Broadcasting an Intent is a background operation that the user is not normally aware of.

Android Broadcast Receiver

UNIVERSITÉ DE NANTES

Type of Broadcasts There are two major classes of broadcasts that can be received:

- **Normal broadcasts** (sent with `sendBroadcast`)
  - are completely asynchronous.
  - All receivers of the broadcast are run in an undefined order, often at the same time.
  - This is more efficient, but means that receivers cannot use the result or abort APIs included here.
- **Ordered broadcasts** (sent with `sendOrderedBroadcast`)
  - are delivered to one receiver at a time.
  - As each receiver executes in turn, it can propagate a result to the next receiver, or it can completely abort the broadcast so that it won't be passed to other receivers.
  - The order receivers run in can be controlled with the `android:priority attribute` of the matching intent-filter; receivers with the same priority will be run in an arbitrary order.

Android Content Provider

- **Content providers** store and retrieve data and make it accessible to *all applications*.
- *They are the only way to share data across Android applications*. There's no common storage area that all Android packages can access.
- Android ships with a number of content providers for common data types (audio, video, images, personal contact information, and so on).

Android Content Provider

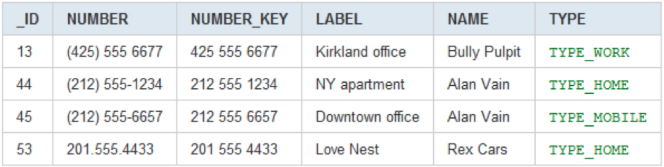
UNIVERSITÉ DE NANTES

- **ContentProviders** are a data layer providing data abstraction for its clients and centralizing storage and retrieval routines in a single place.
- A **ContentProvider** may provide data to an Activity or Service in the same application?s space as well as an Activity or Service contained in other applications.
- A **ContentProvider** may use any form of data storage mechanism available on the Android platform, including files, SQLite databases, or even a memory?based hash map if data persistence is not required.

Android Content Provider

The data model

- **Content providers** expose their data as a simple table on a database model, where each *row* is a record and each *column* is data of a particular type and meaning.
- For example, information about people and their phone numbers might be exposed as follows:



_ID	NUMBER	NUMBER_KEY	LABEL	NAME	TYPE
13	(425) 555 6677	425 555 6677	Kirkland office	Bully Pulpit	TYPE_WORK
44	(212) 555-1234	212 555 1234	NY apartment	Alan Vain	TYPE_HOME
45	(212) 555-6657	212 555 6657	Downtown office	Alan Vain	TYPE_MOBILE
53	201.555.4433	201 555 4433	Love Nest	Rex Cars	TYPE_HOME

Android Content Provider

URIs

- Each content provider exposes a public **URI** that uniquely identifies its data set.
- A content provider that controls multiple data sets (multiple tables) exposes a separate URI for each one.
- All URIs for providers begin with the string `"content://"`.
- Android defines CONTENT_URI constants for all the providers that come with the platform. For example
  - `android.provider.Contacts.Phones.CONTENT_URI` `android.provider.Contacts.Photos.CONTENT_URI`
  - `android.provider.CallLog.Calls.CONTENT_URI` `android.provider.Calendar.CONTENT_URI`
- **The ContentResolver** method takes an URI as its first argument. It's what identifies which provider the ContentResolver should talk to and which table of the provider is being targeted.

Android Manifest xml File

- Every application must have an **AndroidManifest.xml** file (with precisely that name) in its root directory.
- The manifest presents essential information about the application to the Android system, information the system must have before it can run any of the application's code.

Manifest.xml

The manifest does the following:

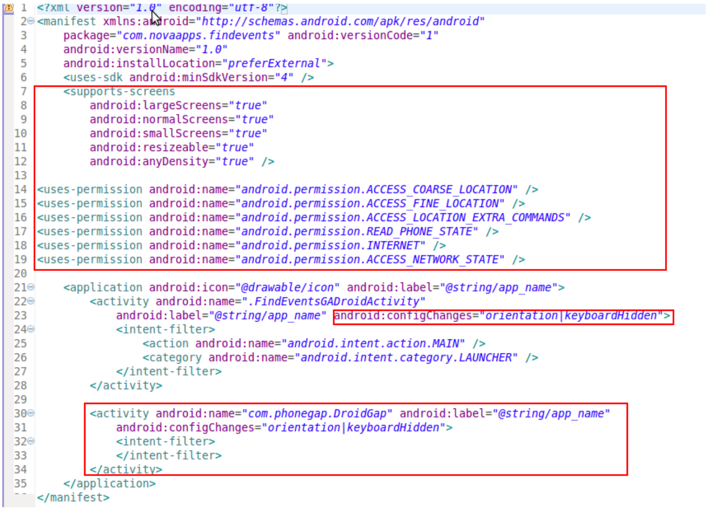
- It names the Java package for the application. The package name serves as a unique identifier for the application.
- It describes the components of the application ? the activities, services, broadcast receivers, and content providers that the application is composed of.
- It determines which processes will host application components.
- It declares which permissions the application must have in order to access protected parts of the API and interact with other applications.
- It also declares the permissions that others are required to have in order to interact with the application's components.
- It lists the Instrumentation classes that provide profiling and other information as the application is running...
- It declares the minimum level of the Android API that the application requires.
- It lists the libraries that the application must be linked against.

Android Manifest xml File

> All the elements that can appear in the manifest file are listed below in alphabetical order.
> These are the only legal elements;
> **you cannot add your own elements or attributes**.

- action
- activity
- activity-alias
- application
- category
- data
- grant-uri-permission
- instrumentation
- intent-filter
- manifest
- meta-data
- permission
- permission-group
- permission-tree
- provider
- receiver
- service
- supports-screens
- uses-configuration
- uses-feature
- uses-library
- uses-permission
- uses-sdk

Manifest.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.novaapps.findevents" android:versionCode="1"
    android:versionName="1.0"
    android:installLocation="preferExternal">
    <uses-sdk android:minSdkVersion="4" />
    <supports-screens
        android:largeScreens="true"
        android:normalScreens="true"
        android:smallScreens="true"
        android:resizeable="true"
        android:anyDensity="true" />

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".FindEventsGADroidActivity"
            android:label="@string/app_name" android:configChanges="orientation|keyboardHidden">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name="com.phonegap.DroidGap" android:label="@string/app_name"
            android:configChanges="orientation|keyboardHidden">
            <intent-filter>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

UNIVERSITÉ DE NANTES

-
-

▶

▶

UNIVERSITÉ DE NANTES

▶

▶

-
-

-
-