

Outils de développement mobile pour le SI de l'entreprise

Introduction à





Présentation

Romain Guidoux

- Promo 2011, F2
- 2011-2015 INRA
- 2015-aj. Braincube





-
- Présentation d'Android
 - A quoi peuvent servir des smartphones dans la recherche & en entreprise ?
 - Présentation de l'environnement de développement
 - Des TDs au fil du cours
 - 2 TPs



Evaluation

- 2 TPs notés (coefs 1 et 2)
 - Retard \leq 48h : -2pts / 24h
 - Supérieur à 48h : 0/20
 - Triche : 0/20
- 1 partiel (coef 3)
 - Pas de code sur feuille
 - Quelques documents autorisés

Présentation d'Android





Qu'est-ce qu'Android ?

- OS mobile
- Développé par l'Open Handset Alliance
- Open source, basé sur un noyau Linux
- Langage de développement : Java
- Pour :
 - Smartphones
 - Tablettes
 - Lunettes (Glass)
 - Montres (Android Wear)
 - TV (Android TV)
 - Voitures (Android Auto)
 - Media players (Nexus Q) ...





Domine le marché mondial

- Part de marché des différents OS

Period	Android	iOS	Windows Phone	Others
2016Q1	83.4%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%
2016Q3	86.8%	12.5%	0.3%	0.4%
2016Q4	81.4%	18.2%	0.2%	0.2%
2017Q1	85.0%	14.7%	0.1%	0.1%

Source: IDC, May 2017



Android... sur quels téléphones ?

- Multitude de smartphones
- Chaque fabricant peut créer un smartphone Android
 - Android Compatibility Definition Document (CDD)

htc

SONY®

acer

lenovo


MOTOROLA

SAMSUNG

 LG



Compatibility Definition Document

- But : Système homogène quel que soit le hardware
- Expose les contraintes matérielles et logicielles
- Exemples :
 - Capteurs : Présence, fréquence, précision, axes
 - Ecran : résolution minimale, ratio (entre 4:3 et 16:9)
 - Stockage
 - Mémoire
 - Connectivité (Wi-Fi, bluetooth...)



Distribution des applications

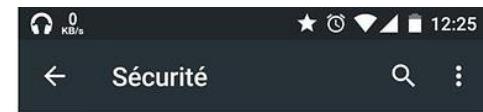
- Google Play Store



- Amazon Apps



- Distribution sur votre site Internet
 - Mise à disposition de l'APK
 - Autoriser l'installation à partir de sources inconnues sur le téléphone



Gestion de l'appareil

Administrateurs de l'appareil
Afficher ou désactiver les administrateurs de l'appareil

Sources inconnues

Autoriser l'installation d'applications issues de sources autres que le Google Play Store

Stockage des identifiants

Type de stockage
Intégré au matériel

Certificats de confiance

Afficher les certificats d'autorité de confiance

Installer depuis la mémoire

Installer les certificats depuis la mémoire de stockage

Gestion d'une application / déploiement sur le Play Store





Google Play Developer Console

- Statistiques très détaillées
- Stack traces des plantages
- Gestion des versions
- Tests automatiques sur différents appareils
- Fiche Play Store

Statistiques

Acquisition d'utilisateurs

Notes et avis

Plantages et ANR

Conseils d'optimisation

Cloud Test Lab

Diagnostics GCM

Fichiers APK

Fiche Google Play Store

Catégorie de contenu

Tarifs et disponibilité

Produits intégrés à l'application

Services et API



Google Play Developer Console

STATISTIQUES

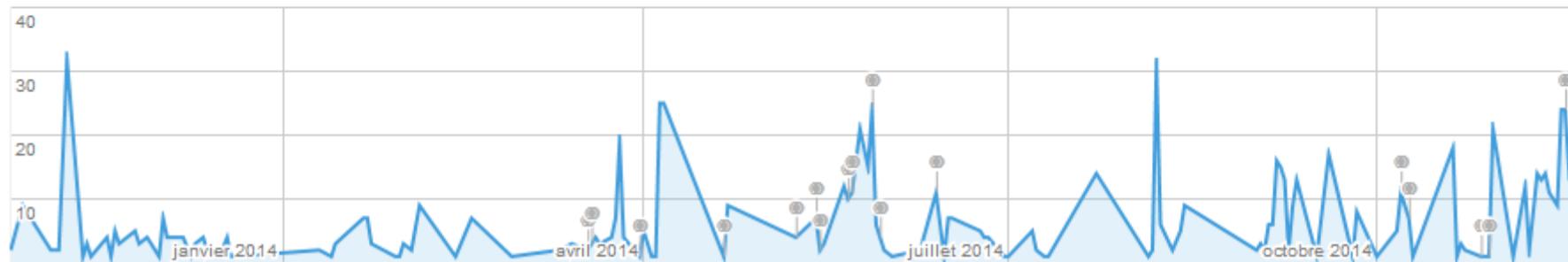
Messages GCM

pour la période 25 oct. 2013 - 19 nov. 2014

Exporter au format CSV

Afficher : le mois dernier 3 mois 6 mois 1 an toutes

Nombre total de messages GCM envoyés à l'application au cours des dernières 24 heures. [En savoir plus](#)



STATISTIQUES

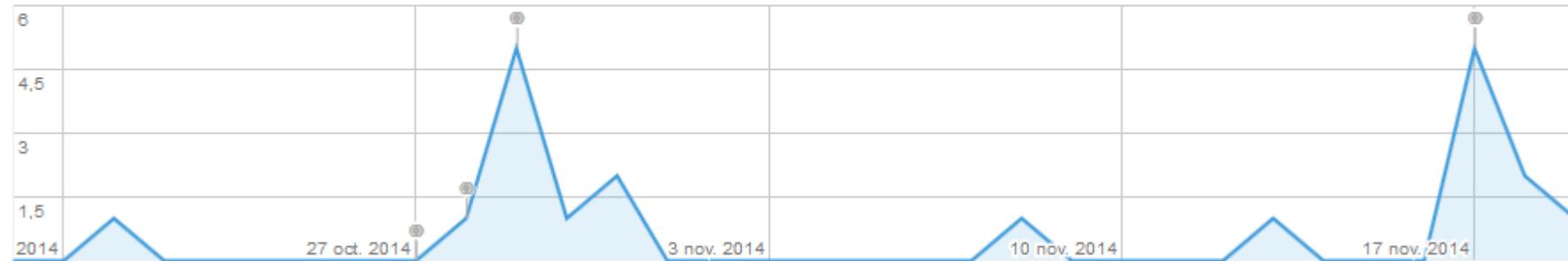
Mises à jour au cours des 24 dernières

pour la période 19 oct. 2014 - 19 nov. 2014

Exporter au format CSV

Afficher : le mois dernier 3 mois 6 mois 1 an toutes

Le nombre d'appareils sur lesquels la version de l'application a été mise à jour au cours des dernières 24 heures. [En savoir plus](#)



Version d'Android

Appareil

Tablettes

Pays

Langue

Version

Opérateur



Google Play Developer Console

FICHIERS APK

[PASSER EN MODE AVANCÉ](#)

PRODUCTION

Version

39

TESTS BÊTA

Configurez des tests bêta pour votre application.

TESTS ALPHA

Configurez des tests alpha pour votre application.

CONFIGURATION DE LA VERSION EN PRODUCTION

[Importer un nouveau fichier APK en version production](#)

FICHIER APK ACTUEL date de publication : 6 janv. 2015 06:46:38

Appareils compatibles

8912[Afficher la liste](#)

Appareils exclus

0[Gérer les appareils exclus](#)

▼ VERSION	IMPORTÉ LE	ÉTAT	ACTIONS
39 (2.6)	6 janv. 2015	en production	

AUTRES FICHIERS APK [Masquer](#)

▼ VERSION	IMPORTÉ LE	ÉTAT	
38 (2.6)	6 janv. 2015	Non publié	Afficher les détails
37 (2.6)	5 janv. 2015	Non publié	Afficher les détails



Publication d'une application

- Engagement juridiquement l'entreprise : autorisation hiérarchique
- Keystore : permet de signer l'application
 - Nécessaire au développeur chargé du build
 - Bien garder le mot de passe
 - Utiliser un keystore par app (ex : vente de l'appli, etc)
 - Sauvegarder !!! (changement de keystore impossible en production)

Perte du keystore = application orpheline

Les différentes versions



Cupcake

Android 1.5



Donut

Android 1.6



Eclair

Android 2.0/2.1



Froyo

Android 2.2/2.2.3



Gingerbread

Android 2.3/2.3.7



Honeycomb

Android 3.0/3.2



Ice Cream Sandwich

Android 4.0/4.0.4



Jelly Bean

Android 4.1/4.3



KitKat

Android 4.4



Lollipop

Android 5.0



Marshmallow

Android 6.0



Android 7.0 Nougat

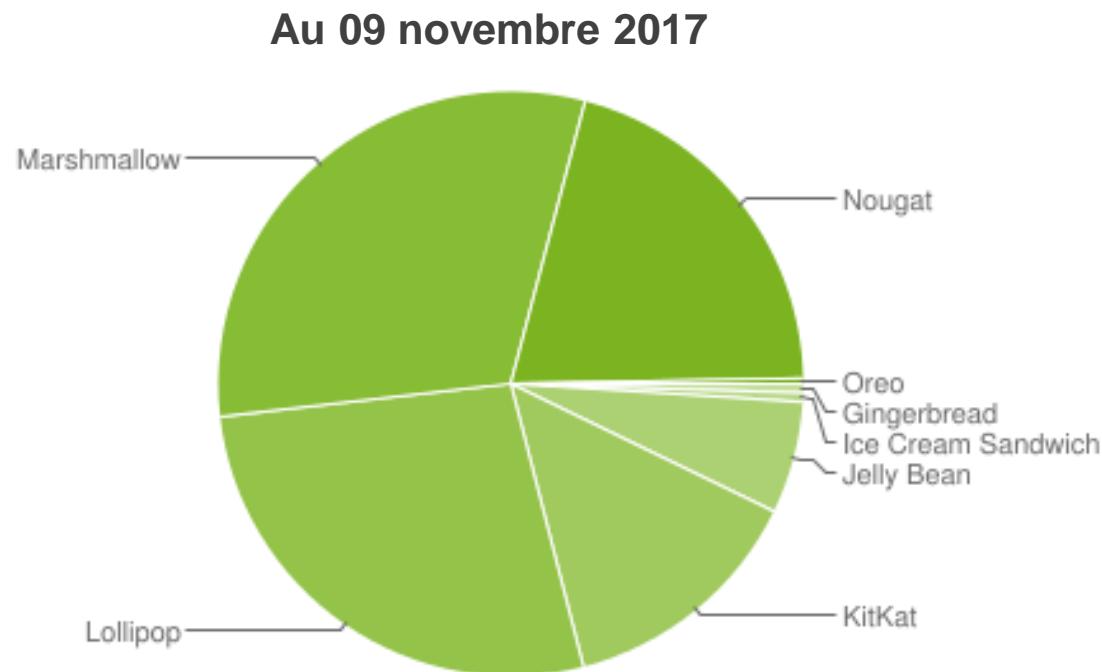


Android 8.0 Oreo



Fragmentation d'Android

- 2017 :
 - **8.0 Oreo**
- 2016 :
 - **7.0 et 7.1 Nougat**
- 2015 :
 - **6.0 Marshmallow**
- 2014 : **5.0 Lollipop**
- 2013 : **4.4 KitKat**
- 2012 : Jelly Bean
- 2011 : ICS, Honeycomb
- 2010 : Froyo, Gingerbread
- 2009 : Cupcake , Donu, Eclair
- 2008 : 1.0



Source : <http://developer.android.com/about/dashboards>



Fragmentation d'Android

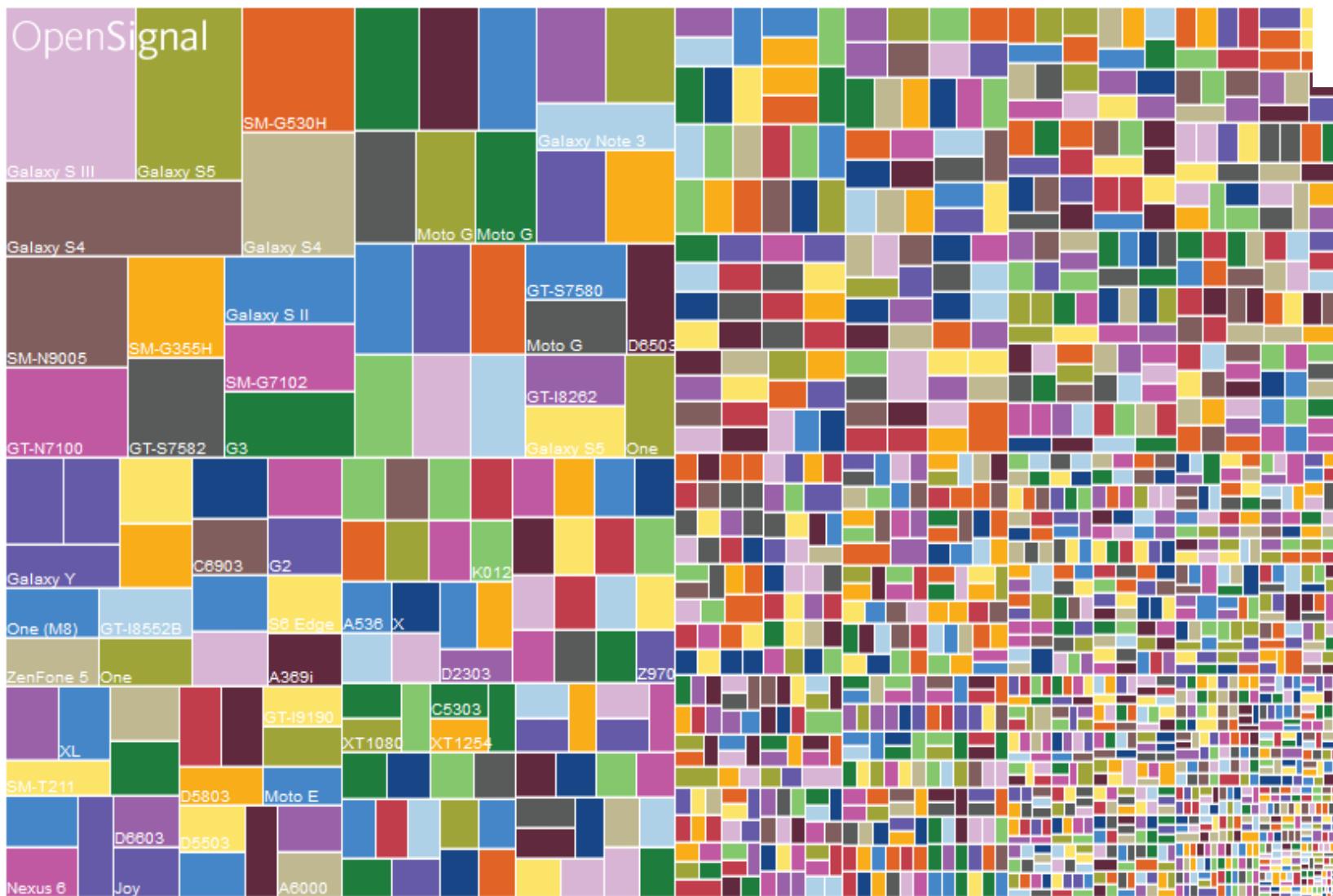
- Fragmentation marquée : 4 versions significatives en circulation
 - Problème lors de la conception : supporter anciennes et nouvelles versions
- Effort de Google sur l'Android Support Library



Fragmentation d'Android



DEVICE FRAGMENTATION



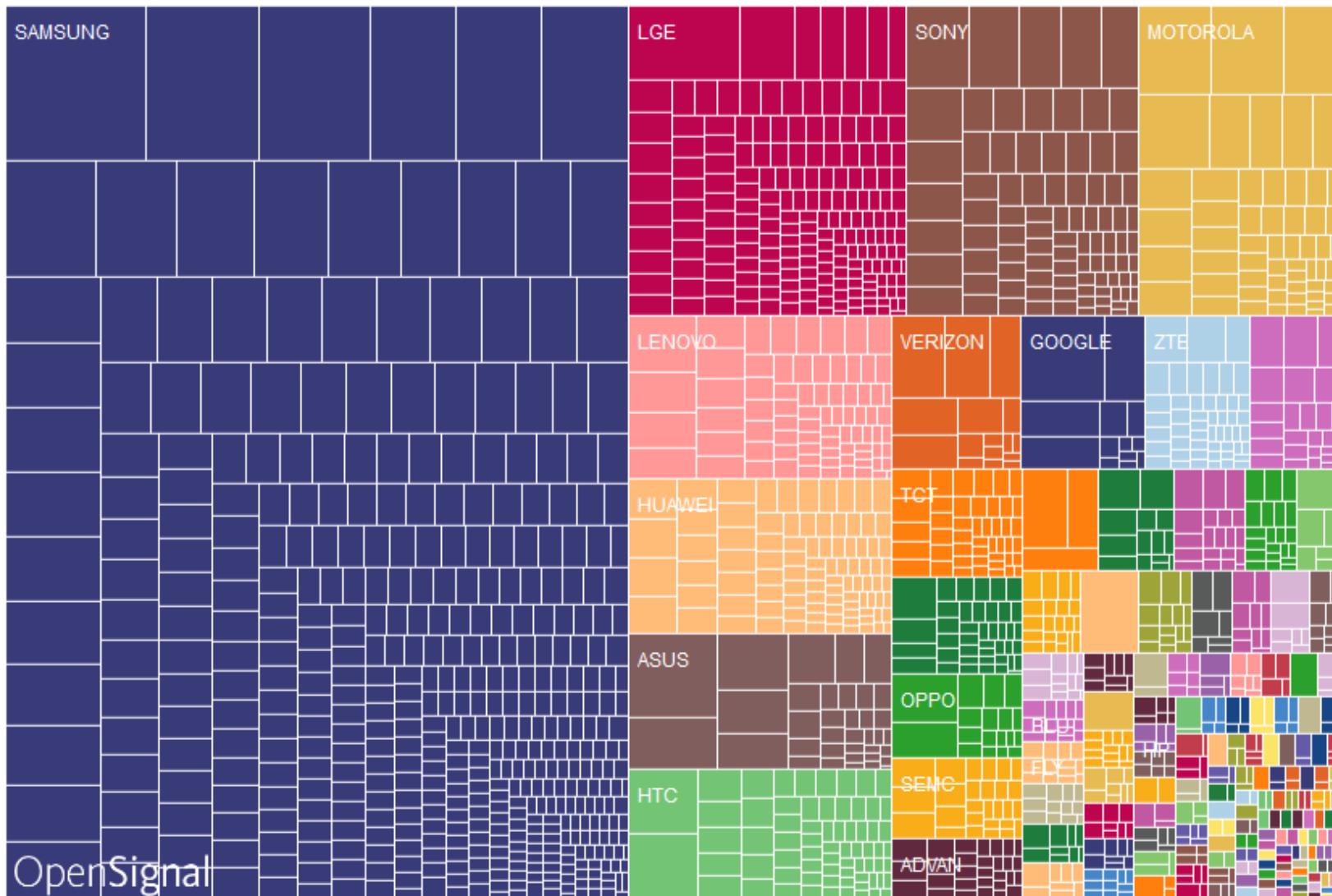
Source : OpenSignal <http://goo.gl/gpRVDj>



Fragmentation d'Android



BRAND FRAGMENTATION



Source : OpenSignal <http://goo.gl/gpRVDj>



Fragmentation d'Android



- Une fragmentation de devices qui s'accentue

24,093

Août 2014 – Août 2015 Distinct Android devices seen this year

18,796

Août 2013 – Août 2014 Distinct Android devices seen last year

682,000

Devices surveyed for this report.

37.8%

Samsung's share of those devices.

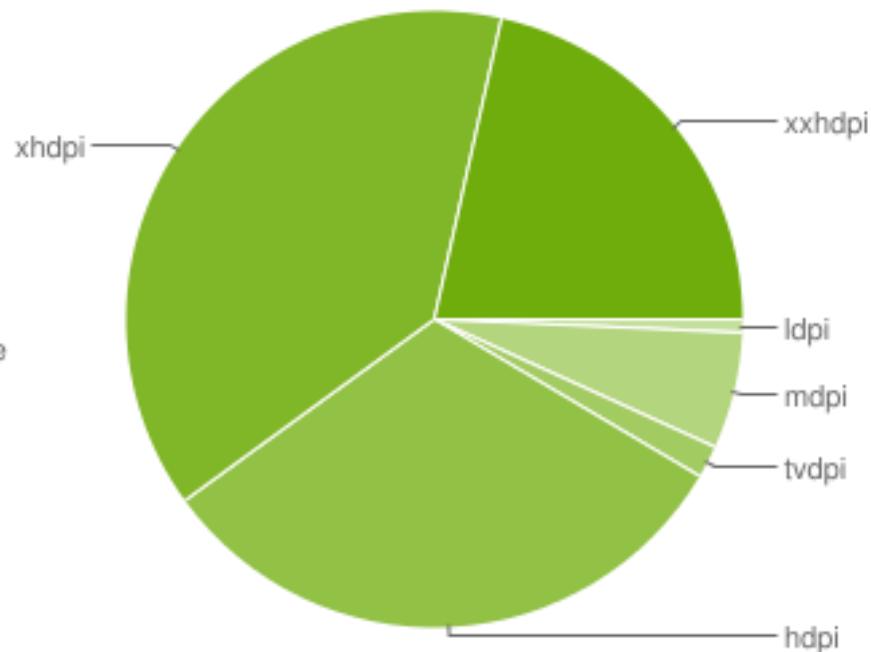
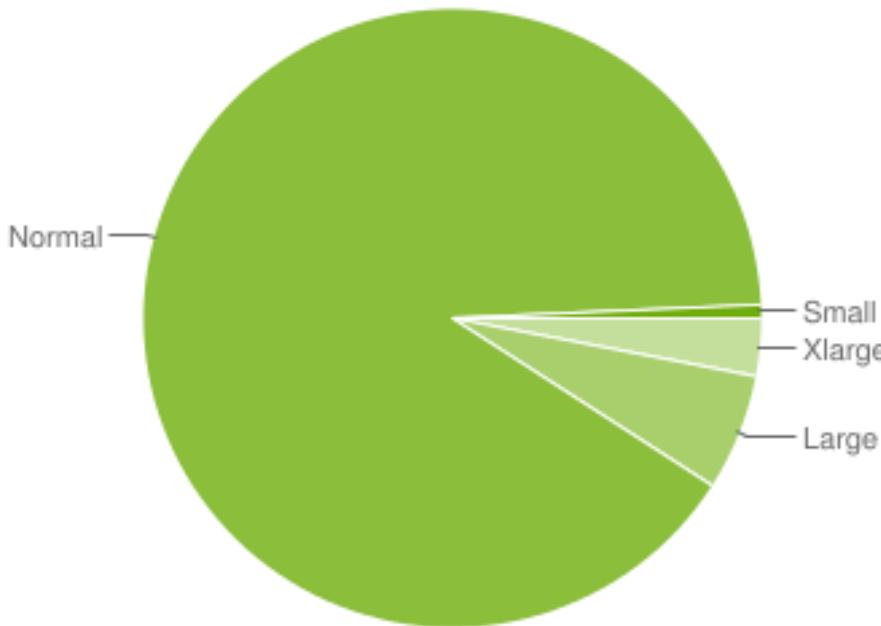
1,294

Device brands seen this year



Fragmentation d'Android

- Taille d'écran et densité d'affichage



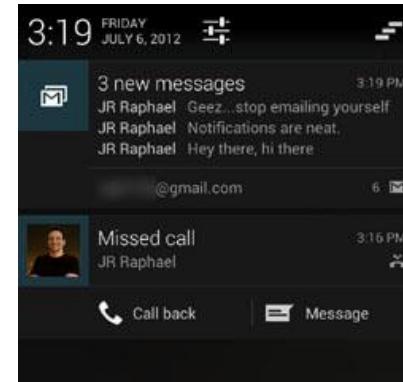
Au 09 novembre 2017



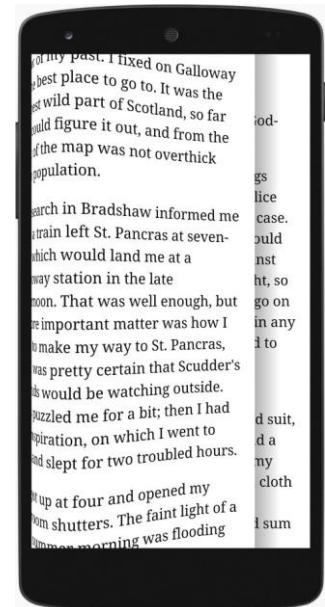
Versions

- Honeycomb (3.0, API 11) : réservée aux tablettes
- ICS (4.0, API 14) : version unifiée pour smartphones et tablettes

- Jelly Bean :
 - Notifications améliorées
 - Effort sur la fluidité du système
 - Multi-comptes (tablettes)



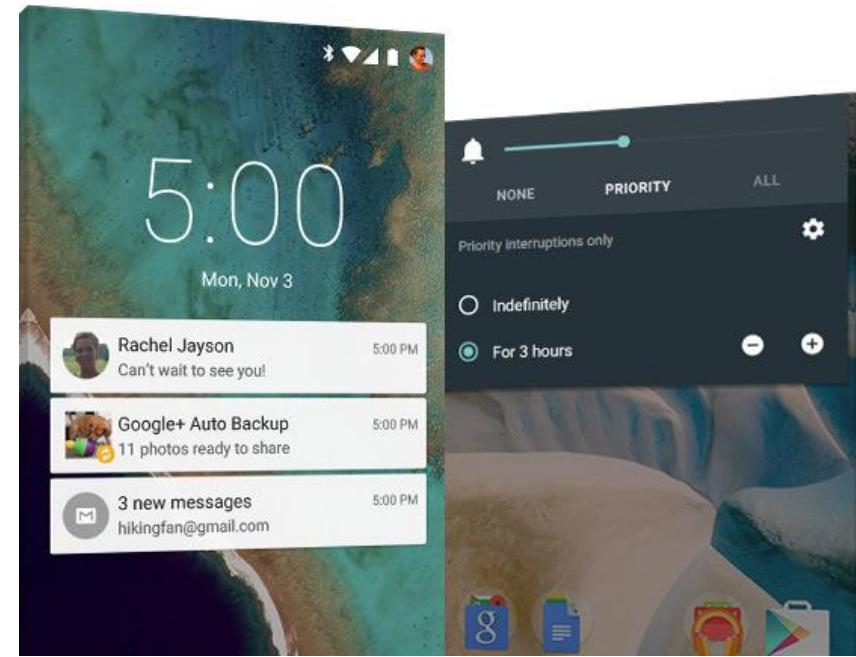
- KitKat
 - Consomme moins de RAM
 - Bluetooth 4.0
 - Sensor batching





Versions

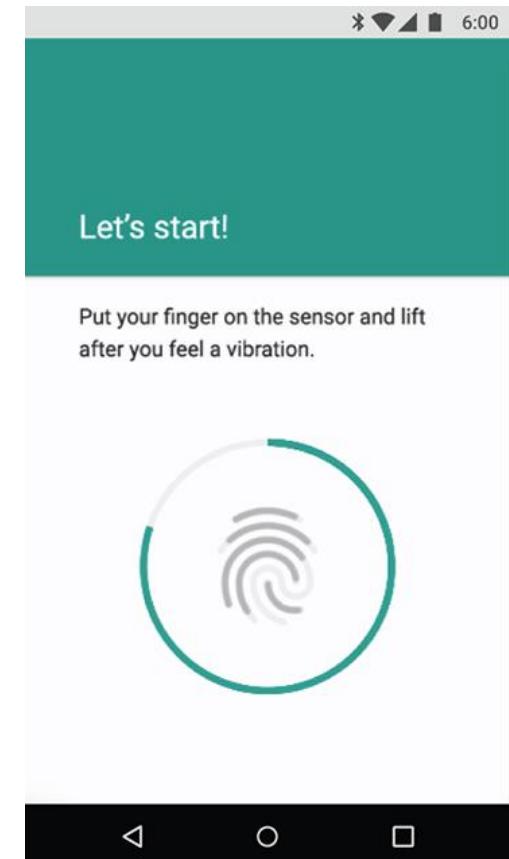
- Honeycomb (3.0, API 11) : réservée aux tablettes
- ICS (4.0, API 14) : version unifiée pour smartphones et tablettes
- Jelly Bean
- KitKat
- Lollipop :
 - Consommation batterie
 - Notifications améliorées
 - Material Design
 - Device encryption par défaut
 - Compte invité, multicomptes smartphones





Versions

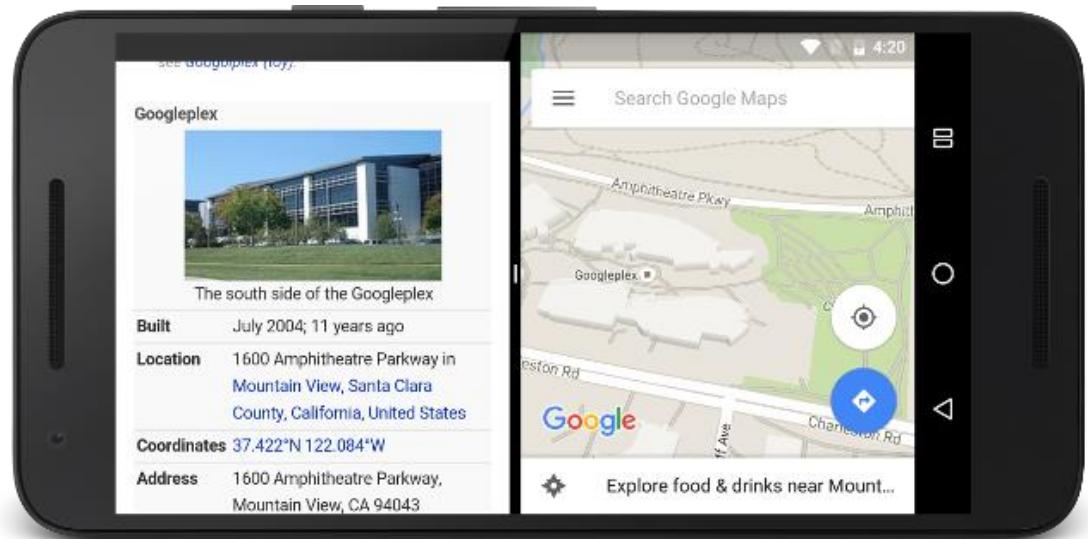
- Honeycomb (3.0, API 11) : réservée aux tablettes
- ICS (4.0, API 14) : version unifiée pour smartphones et tablettes
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow
 - Meilleure gestion de la batterie
 - Authentification empreintes digitales
 - Support USB Type C
 - Refonte de la gestion des permissions





Versions

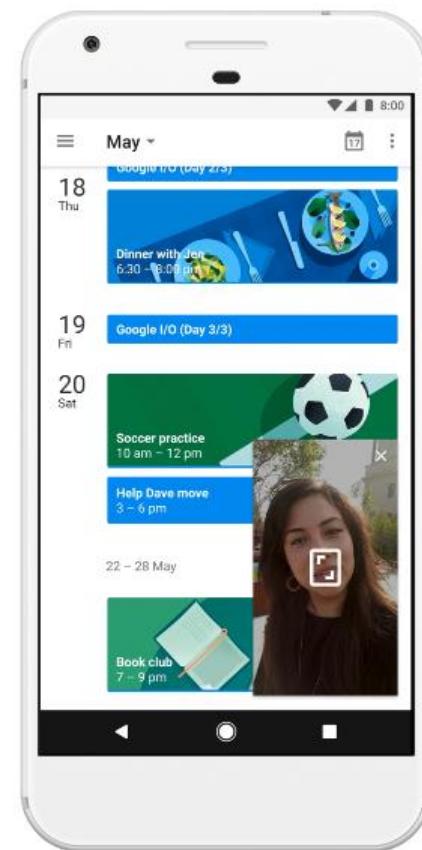
- Honeycomb (3.0, API 11) : réservée aux tablettes
- ICS (4.0, API 14) : version unifiée pour smartphones et tablettes
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow
- Nougat
 - Multifénêtrage
 - Mise à jour du système transparente





Versions

- Honeycomb (3.0, API 11) : réservée aux tablettes
- ICS (4.0, API 14) : version unifiée pour smartphones et tablettes
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow
- Nougat
- Oreo
 - Boot plus rapide
 - Gestion plus stricte des tâches de fond
 - Picture-in-picture



Utilisation des smartphones dans la recherche et en entreprise





Projet



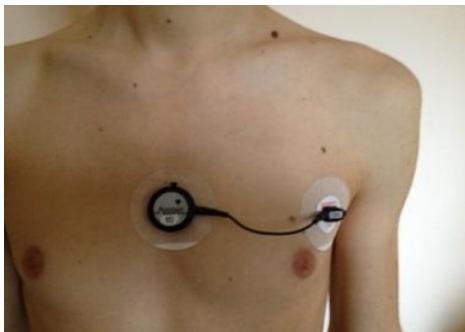
- Estimation de la **dépense énergétique** à partir des données d'accélérométrie collectées par des **smartphones**
- Projet INRA, Almerys, CHU G. Montpied
- Possibilité de **stages**





Dans la recherche

- Estimation de l'intensité des activités physiques et de la dépense énergétique avec un smartphone Android
- Avantages pour l'INRA :
 - Se passer des capteurs de recherche
 - Coût
 - Eloignement des volontaires
 - Traitement automatisé des données





Principe de fonctionnement

Affichage pour visualisation par le volontaire





Utilisation en entreprise

- Exemples d'utilisation :
 - Consulter / modifier le stock sur son smartphone
 - Accéder aux dossiers clients
 - Consulter / modifier une fiche d'intervention
 - Scan de codes barres
 - Facturation mobile
 - Localiser une flotte de vendeurs / de camions / ...
 - ...



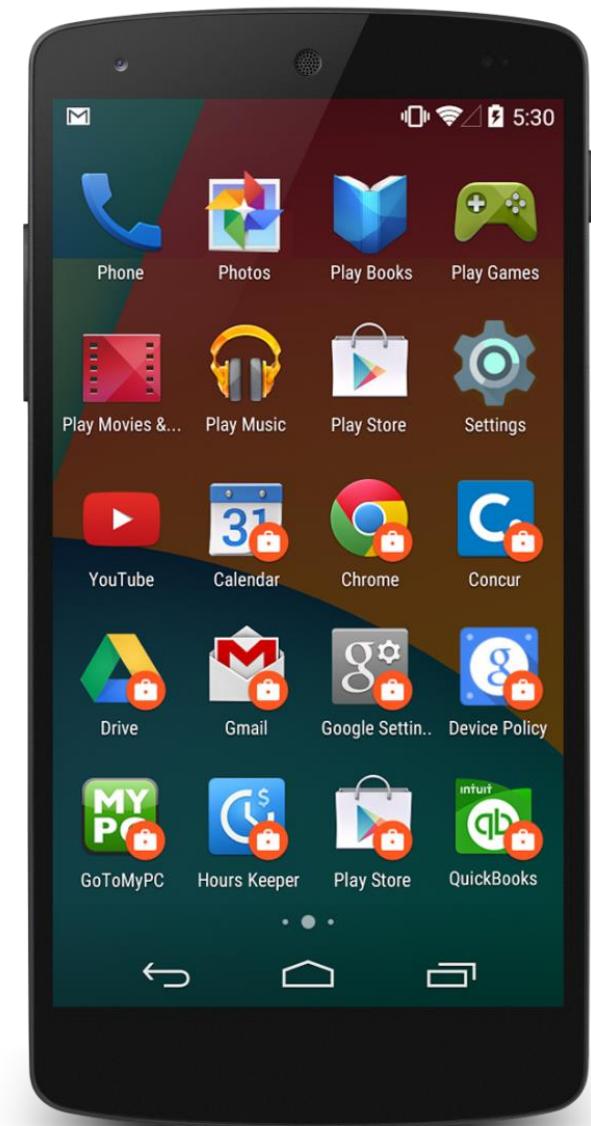
Utilisation en entreprise

- Questions à se poser :
 - Déploiement via les App Stores officiels ou en interne ?
 - Autoriser les employés à :
 - Utiliser leurs propres appareils ? (BYOD)
 - Utiliser leurs propres applications ? (BYOA)
 - Gérer la diversité des équipements mobiles : Android, iOS, Windows Phone
 - Sécuriser les terminaux (distinction données perso / pro, chiffrement)
 - Que faire en cas de vol / perte / piratage ?



Utilisation en entreprise

- Android for Work : Séparation des données perso & pro
- Déploiement de lots d'applications par l'entreprise
 - « Managed apps »
- https://www.android.com/intl/fr_fr/work



Environnement de travail





Quel IDE choisir ?

- **Android Studio**
 - Basé sur IntelliJ IDEA Community
 - Développé par Google





Vocabulaire IntelliJ IDEA

- Attention au vocabulaire != Eclipse

Eclipse	IDEA
Workspace	Project
Project	Module
Project-specific JRE	Module JDK
User library	Global library
Classpath variable	Path variable
Project dependency	Module dependency
Library	Module library

Environnement de travail

TD : Configuration de l'environnement





Installation de Java

1. Vérifiez que Java est installé

- Si ce n'est pas le cas :
 - Rendez-vous sur le site ninite.com
 - Cochez Java 8 et JDK 8
 - Get Installer

Runtimes

- Java 8
- .NET 4.6.1
- Silverlight
- Air
- Shockwave

Developer Tools

- Python
- FileZilla
- Notepad++
- JDK 8
- JDK x64 8
- WinSCP
- PuTTY
- WinMerge
- Eclipse



Installation du bundle

2. Téléchargez le bundle Android Studio + Android SDK

<https://developer.android.com/sdk/index.html>

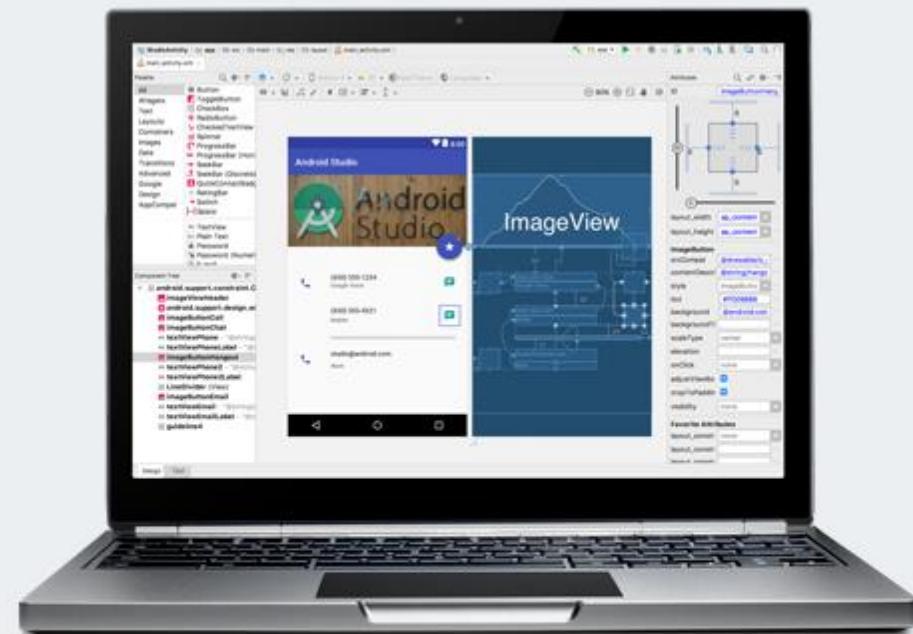
Android Studio

The Official IDE for Android

Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

**DOWNLOAD ANDROID STUDIO
3.0 FOR WINDOWS (681 MB)**





Installation du bundle

Installing Android Studio

Android Studio provides everything you need to start developing apps for Android, including the Android Studio IDE and the Android SDK tools.

If you didn't download Android Studio, go [download Android Studio now](#), or switch to the [stand-alone SDK Tools install](#) instructions.

Before you set up Android Studio, be sure you have installed JDK 6 or higher (the JRE alone is not sufficient)—JDK 7 is required when developing for Android 5.0 and higher. To check if you have JDK installed (and which version), open a terminal and type `javac -version`. If the JDK is not available or the version is lower than version 6, download the [Java SE Development Kit 7](#).

To set up Android Studio on Windows:

[\[Show instructions for all platforms \]](#)

1. Launch the `.exe` file you just downloaded.
2. Follow the setup wizard to install Android Studio and any necessary SDK tools.

On some Windows systems, the launcher script does not find where Java is installed. If you encounter this problem, you need to set an environment variable indicating the correct location.

Select **Start menu > Computer > System Properties > Advanced System Properties**. Then open **Advanced tab > Environment Variables** and add a new system variable `JAVA_HOME` that points to your JDK folder, for example
`C:\Program Files\Java\jdk1.7.0_21`.

The individual tools and other SDK packages are saved outside the Android Studio application directory. If you need to access the tools directly, use a terminal to navigate to the location where they are installed. For example:

```
\Users\<user>\sdk\
```

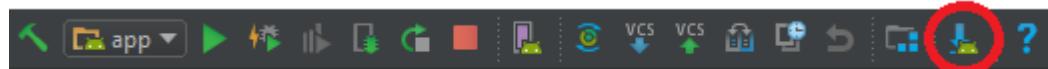
Android Studio is now ready and loaded with the Android developer tools, but there are still a couple packages you should add to make your Android SDK complete.



SDK Manager

3. Lancez l'Android SDK Manager. Pour cela :

- Ouvrez Android Studio en tant qu'administrateur
- Vérifiez que la toolbar est visible (View > Toolbar)
- Dans la toolbar, ouvrez le SDK Manager





Modules du SDK

- Versions d'Android (les API) :

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
►	<input type="checkbox"/> Android API 27			
►	<input type="checkbox"/> Android 8.0 (Oreo)			
►	<input type="checkbox"/> Android 7.1.1 (Nougat)			
▼	<input checked="" type="checkbox"/> - Android 7.0 (Nougat)			
	<input checked="" type="checkbox"/> Google APIs, Android 24	24	1.0.0	Installed
	<input checked="" type="checkbox"/> Android SDK Platform 24, rev 2	24	2	Installed
	<input checked="" type="checkbox"/> Sources for Android 24	24	1	Installed
	<input type="checkbox"/> Android TV Intel x86 Atom System Image	24	14	Not installed
	<input type="checkbox"/> ARM 64 v8a System Image	24	7	Not installed
	<input type="checkbox"/> ARM EABI v7a System Image	24	7	Not installed
	<input checked="" type="checkbox"/> Intel x86 Atom System Image	24	8	Installed
	<input checked="" type="checkbox"/> Intel x86 Atom_64 System Image	24	8	Installed
	<input type="checkbox"/> Google APIs ARM 64 v8a System Image	24	19	Not installed
	<input type="checkbox"/> Google APIs ARM EABI v7a System Image	24	19	Not installed
	<input checked="" type="checkbox"/> Google APIs Intel x86 Atom System Image	24	19	Installed
	<input checked="" type="checkbox"/> Google APIs Intel x86 Atom_64 System Image	24	19	Installed
	<input type="checkbox"/> Google Play Intel x86 Atom System Image	24	19	Not installed
▼	<input type="checkbox"/> - Android 6.0 (Marshmallow)			
	<input type="checkbox"/> Google APIs	23	1	Not installed
	<input type="checkbox"/> Android SDK Platform 23	23	3	Not installed
	<input type="checkbox"/> Sources for Android 23	23	1	Not installed
	<input type="checkbox"/> Android TV ARM EABI v7a System Image	23	12	Not installed
	<input type="checkbox"/> Android TV Intel x86 Atom System Image	23	13	Not installed

Show Package Details



Modules du SDK

- SDK Tools :
 - Emulateur
 - DDMS (Dalvik Debug Monitor Server) : pour récupérer des infos lors du debug, faire des captures d'écran...
 - ...

Name	Version	Status
Android SDK Build-Tools		Installed
GPU Debugging tools		Not Installed
CMake		Not Installed
LLDB		Not Installed
Android Auto API Simulators	1	Not installed
Android Auto Desktop Head Unit emulator	1.1	Not installed
Android Emulator	26.1.4	Installed
Android SDK Platform-Tools	26.0.2	Installed
Android SDK Tools	26.1.1	Installed
Android Support Library, rev 23.2.1	23.2.1	Installed
Documentation for Android SDK	1	Installed
GPU Debugging tools, rev 1.0.3	1.0.3	Installed
Google Play APK Expansion library	1	Not installed
Google Play Licensing Library	1	Not installed
Google Play services	46	Installed
Google USB Driver, rev 11	11.0.0	Installed
Google Web Driver	2	Not installed
Instant Apps Development SDK	1.1.0	Installed
Intel x86 Emulator Accelerator (HAXM installer)	6.2.1	Installed
NDK	16.0.4442984	Not installed
Support Repository		

Show Package Details



Modules du SDK

- Android SDK Platform-tools :
 - ADB (Android Debug Bridge) : script qui permet la communication avec un émulateur ou un device
 - ...

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates.
Check "show package details" to display available versions of an SDK Tool.

	Name	Version	Status
<input checked="" type="checkbox"/>	Android SDK Build-Tools		Installed
<input type="checkbox"/>	GPU Debugging tools		Not Installed
<input type="checkbox"/>	CMake		Not Installed
<input type="checkbox"/>	LLDB		Not Installed
<input type="checkbox"/>	Android Auto API Simulators	1	Not installed
<input type="checkbox"/>	Android Auto Desktop Head Unit emulator	1.1	Not installed
<input checked="" type="checkbox"/>	Android Emulator	26.1.4	Installed
<input checked="" type="checkbox"/>	Android SDK Platform-Tools	26.0.2	Installed
<input checked="" type="checkbox"/>	Android SDK Tools	26.1.1	Installed
<input checked="" type="checkbox"/>	Android Support Library, rev 23.2.1	23.2.1	Installed
<input checked="" type="checkbox"/>	Documentation for Android SDK	1	Installed
<input checked="" type="checkbox"/>	GPU Debugging tools, rev 1.0.3	1.0.3	Installed
<input type="checkbox"/>	Google Play APK Expansion library	1	Not installed
<input type="checkbox"/>	Google Play Licensing Library	1	Not installed
<input checked="" type="checkbox"/>	Google Play services	46	Installed
<input checked="" type="checkbox"/>	Google USB Driver, rev 11	11.0.0	Installed
<input type="checkbox"/>	Google Web Driver	2	Not installed
<input checked="" type="checkbox"/>	Instant Apps Development SDK	1.1.0	Installed
<input checked="" type="checkbox"/>	Intel x86 Emulator Accelerator (HAXM installer)	6.2.1	Installed
<input type="checkbox"/>	NDK	16.0.4442984	Not installed
<input checked="" type="checkbox"/>	Support Repository		

Show Package Details



Modules du SDK

- Android SDK Build tools:
 - dexdump : conversion des fichiers .dex en bytecode
 - zipalign : optimisation des fichiers APK
 - ...

SDK Platforms **SDK Tools** SDK Update Sites

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

Name	Version	Status
<input checked="" type="checkbox"/> Android SDK Build-Tools		Installed
<input type="checkbox"/> GPU Debugging tools		Not Installed
<input type="checkbox"/> CMake		Not Installed
<input type="checkbox"/> LLDB		Not Installed
<input type="checkbox"/> Android Auto API Simulators	1	Not installed
<input type="checkbox"/> Android Auto Desktop Head Unit emulator	1.1	Not installed
<input checked="" type="checkbox"/> Android Emulator	26.1.4	Installed
<input checked="" type="checkbox"/> Android SDK Platform-Tools	26.0.2	Installed
<input checked="" type="checkbox"/> Android SDK Tools	26.1.1	Installed
<input checked="" type="checkbox"/> Android Support Library, rev 23.2.1	23.2.1	Installed
<input checked="" type="checkbox"/> Documentation for Android SDK	1	Installed
<input checked="" type="checkbox"/> GPU Debugging tools, rev 1.0.3	1.0.3	Installed
<input type="checkbox"/> Google Play APK Expansion library	1	Not installed
<input type="checkbox"/> Google Play Licensing Library	1	Not installed
<input checked="" type="checkbox"/> Google Play services	46	Installed
<input checked="" type="checkbox"/> Google USB Driver, rev 11	11.0.0	Installed
<input type="checkbox"/> Google Web Driver	2	Not installed
<input checked="" type="checkbox"/> Instant Apps Development SDK	1.1.0	Installed
<input checked="" type="checkbox"/> Intel x86 Emulator Accelerator (HAXM installer)	6.2.1	Installed
<input type="checkbox"/> NDK	16.0.4442984	Not installed
► <input checked="" type="checkbox"/> Support Repository		

Show Package Details

Peut installer 1 ou plusieurs versions

Version utilisée déclarée dans le projet



Modules du SDK

- Les extras :

- Android Support Library : permet de porter des fonctionnalités récentes sur d'anciennes versions d'Android
- Google Play Services : services Google (Maps, Drive, Sign-In...)

SDK Platforms **SDK Tools** SDK Update Sites

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates.
Check "show package details" to display available versions of an SDK Tool.

	Name	Version	Status
<input checked="" type="checkbox"/>	Android SDK Build-Tools		Installed
<input type="checkbox"/>	GPU Debugging tools		Not Installed
<input type="checkbox"/>	CMake		Not Installed
<input type="checkbox"/>	LLDB		Not Installed
<input type="checkbox"/>	Android Auto API Simulators	1	Not installed
<input type="checkbox"/>	Android Auto Desktop Head Unit emulator	1.1	Not installed
<input checked="" type="checkbox"/>	Android Emulator	26.1.4	Installed
<input checked="" type="checkbox"/>	Android SDK Platform-Tools	26.0.2	Installed
<input checked="" type="checkbox"/>	Android SDK Tools	26.1.1	Installed
<input checked="" type="checkbox"/>	Android Support Library, rev 23.2.1	23.2.1	Installed
<input checked="" type="checkbox"/>	Documentation for Android SDK	1	Installed
<input checked="" type="checkbox"/>	GPU Debugging tools, rev 1.0.3	1.0.3	Installed
<input type="checkbox"/>	Google Play APK Expansion library	1	Not installed
<input type="checkbox"/>	Google Play Licensing Library	1	Not installed
<input checked="" type="checkbox"/>	Google Play services	46	Installed
<input checked="" type="checkbox"/>	Google USB Driver, rev 11	11.0.0	Installed
<input type="checkbox"/>	Google Web Driver	2	Not installed
<input checked="" type="checkbox"/>	Instant Apps Development SDK	1.1.0	Installed
<input checked="" type="checkbox"/>	Intel x86 Emulator Accelerator (HAXM installer)	6.2.1	Installed
<input type="checkbox"/>	NDK	16.0.4442984	Not installed
<input checked="" type="checkbox"/>	Support Repository		

Show Package Details



Modules du SDK

4. Cochez et installez / mettez à jour :

- Android SDK Tools
- Android SDK Platform Tools
- Android SDK Build Tools le plus récent
- Android 6.0 (API 23) ou 7.0 (API 24)
 - SDK
 - Google APIs
 - Google APIs x86
 - Google APIs x86 64bits
 - Intel x86
 - Intel x86 64bits
- Android Support Library
- Google Play Services
- Google USB Driver
- Intel x86 HAXM installer



Installation de HAXM

5. Allez dans le dossier de votre SDK

C:\Users\username\AppData\Local\Android\sdk\
extras\intel\Hardware_Accelerated_Execution_Manager

Nom	Modifié le	Type	Taille
intelhaxm-android.exe	25/10/2015 13:06	Application	1 996 Ko
Release Notes.txt	25/10/2015 13:06	Fichier TXT	6 Ko
silent_install.bat	25/10/2015 13:06	Fichier de comma...	10 Ko
silent_install_readme.txt	25/10/2015 13:06	Fichier TXT	3 Ko
source.properties	25/10/2015 13:06	Fichier PROPERTIES	3 Ko



A quoi ça sert ?

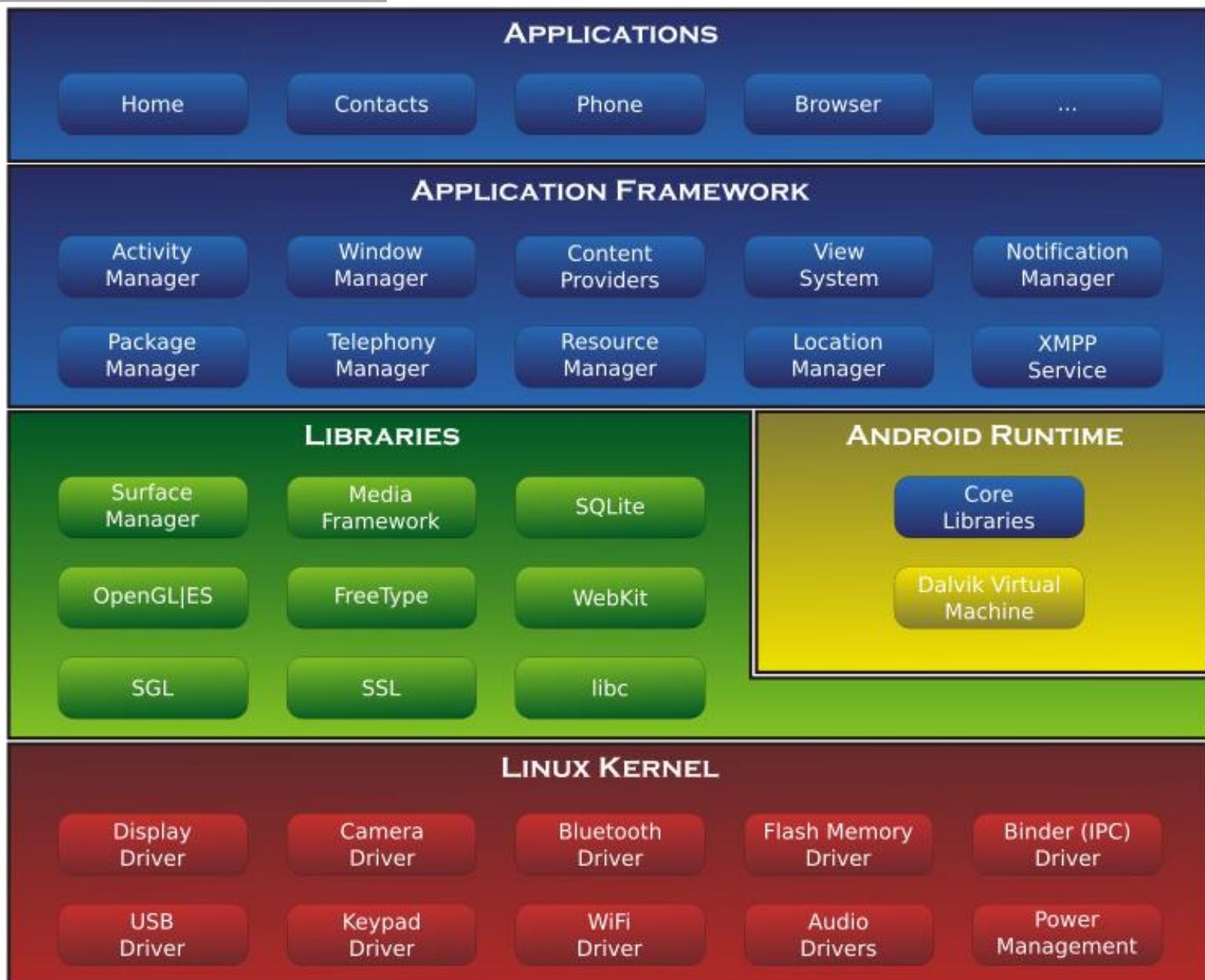
- Performances de l'émulateur
- S'utilise en combinaison avec les images x86 de l'émulateur Android

Architecture du système





Architecture du système





Noyau Linux



- Noyau Linux profondément modifié
 - Version 2.6
 - Version 3.x depuis Android 4.0
- Drivers audio, USB, bluetooth...
- Gestion de l'énergie adaptée aux mobiles



Bibliothèques



- Bibliothèques écrites en C
 - Bases de données (SQLite)
 - Graphique (OpenGL)
 - Rendu de pages web (WebKit)
 - Gestion des formats audio et vidéo (Media Framework)
 - ...



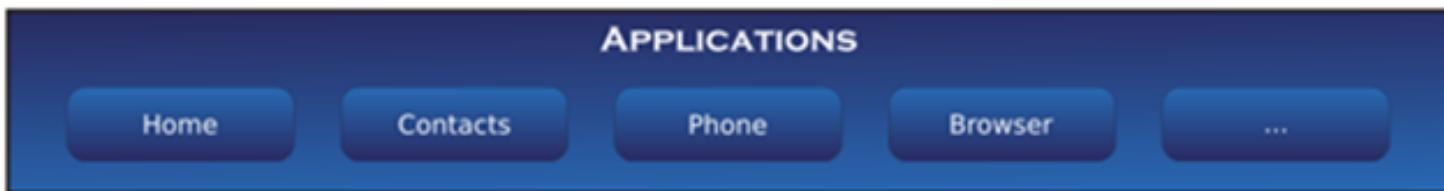
Application Framework



- Eléments de gestion utilisables dans vos applications
`getSystemService(Context.TELEPHONY_SERVICE)`
- **Activity Manager** : gère la pile de fenêtres (Activities) et leur cycle de vie
- **Resource Manager** : gestion des chaînes de caractères (i18n), des images, des layouts...
- **Telephony Manager** : Informations sur l'état des services de téléphonie (2G, 3G, 4G, WiFi, data ?)
- **View System** : widgets (boutons, listes, spinners...), événements graphiques (clicks, touch...) et layouts



Applications



- Applications par défaut :
 - Home : écran principal
 - Contacts
 - Navigateur Internet
 - Téléphone
 - ...



Notions de base

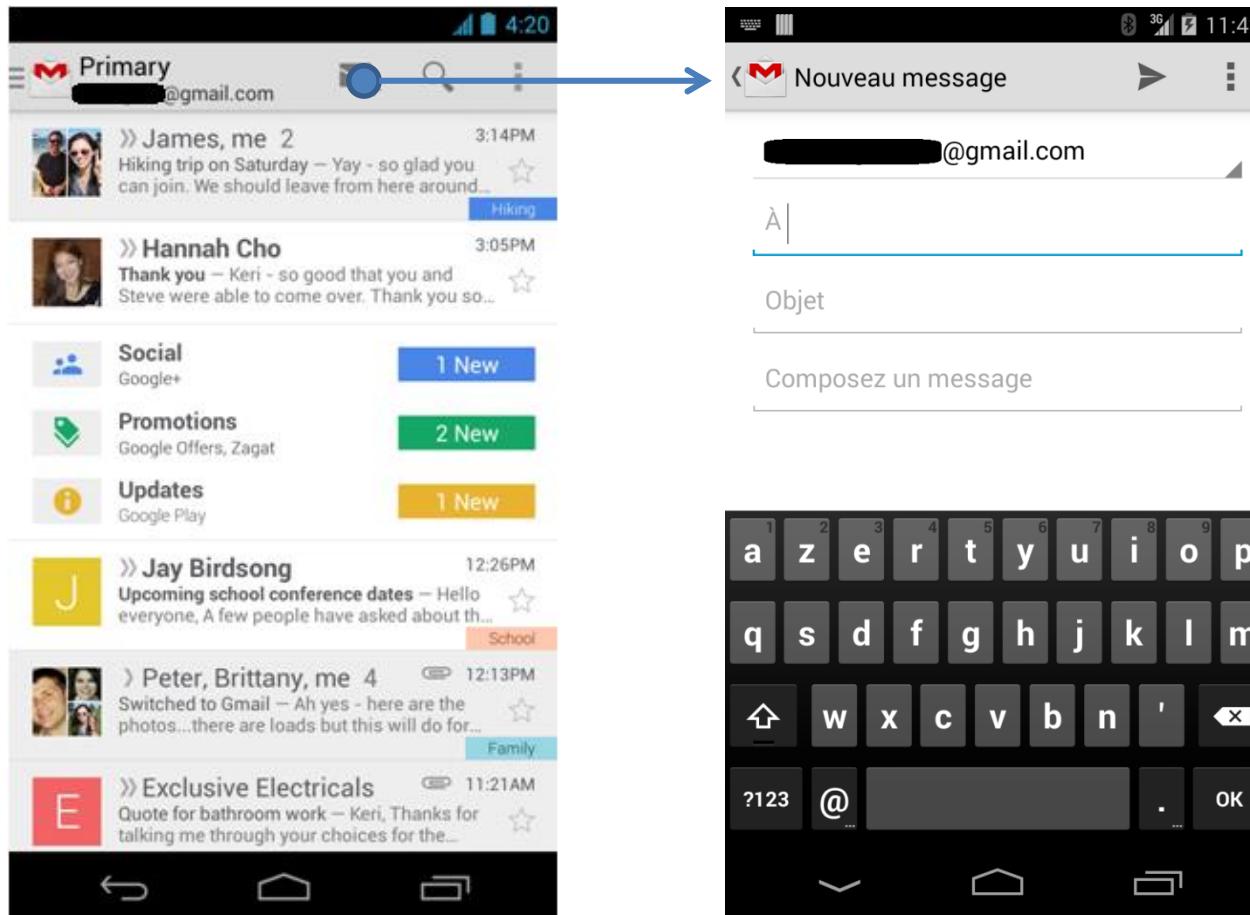




Un peu de vocabulaire

- **Activity** ⇔ Fenêtre graphique

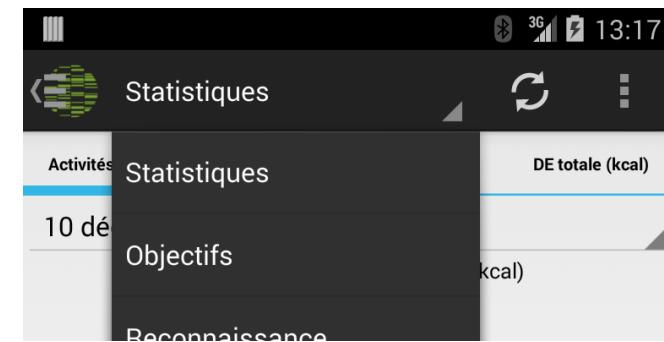
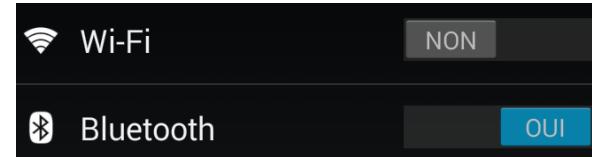
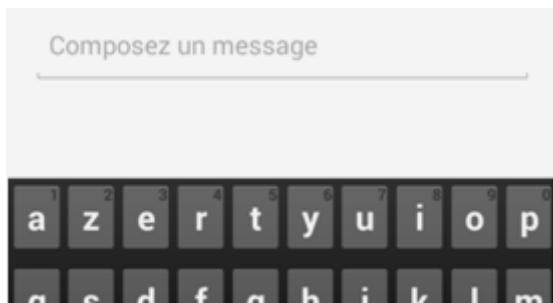
« An activity is a single, focused thing that the user can do. »





Un peu de vocabulaire

- Activity
- **Layout et Views**
 - Types de vues : EditText, TextView, Spinner, Switch, List...
 - Layout :
 - Ensemble de vues
 - Fichier XML





Un peu de vocabulaire

- Activity
- Layout et Views
- **Service :**
 - Permet d'effectuer des opérations en background
 - Pas d'interface utilisateur
 - Peut fonctionner même après que l'utilisateur ait quitté l'appli
 - Peut être mis à disposition d'autres applications
 - Exemples : lecteur de musique, collecte et envoi de données via Internet...



Un peu de vocabulaire

- Activity
- Layout et Views
- Service
- Intent :
 - Messages pour une communication entre composants
 - Peut contenir des données
 - Implicite ou explicite

```
Intent intent = new Intent(Intent.ACTION_VIEW)
intent.setData(Uri.parse("http://www.google.fr"));
startActivity(intent);
```

```
startActivity(new Intent(this, ActivityB.class));
startService(new Intent(this, MonService.class));
```



Un peu de vocabulaire

- Activity
- Layout et Views
- Service
- Intent
- **Broadcast Receiver :**
 - Composant qui permet de répondre à un appel envoyé par broadcast (càd : à tout le monde)
 - Beaucoup utilisé par le système
 - Batterie faible
 - Mémoire faible
 - Boot
 - Appel entrant
 - ...
 - Les apps peuvent définir et envoyer leurs propres broadcasts



Un peu de vocabulaire

- Activity
- Layout et Views
- Service
- Intent
- Broadcast Receiver
- **Content Provider :**
 - Gère des données liées à l'application
 - Différentes sources : BDD, fichiers, Internet...
 - URI commence par content://
 - Peut autoriser les autres apps à lire et modifier les données



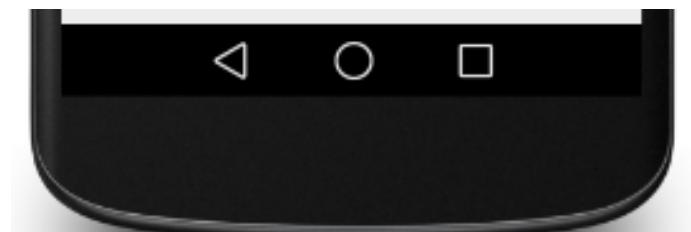
Un peu de vocabulaire

- Activity
- Layout et Views
- Service
- Intent
- Broadcast Receiver
- Content Provider
- **Thread :**
 - Main Thread : Gère les événements, la construction des widgets...
 - Longue opération dans le main thread = interface gelée = utilisateur mécontent
 - Classe Thread, AsyncTask

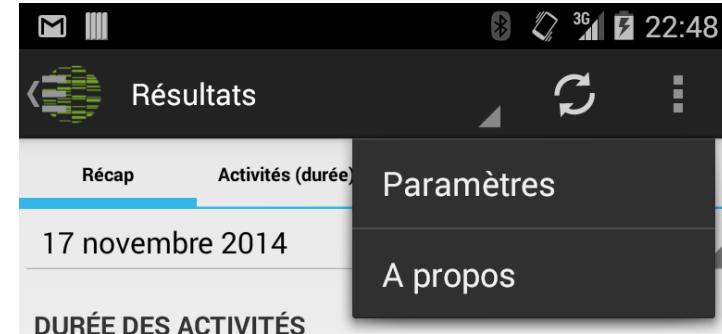
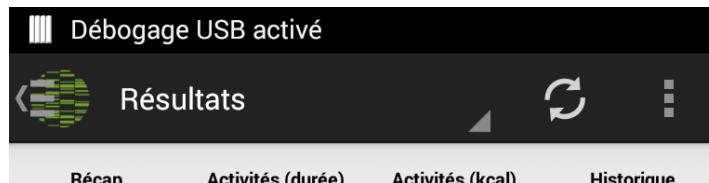


Boutons de base

- Back : précédent
- Home : retour à l'application... Home

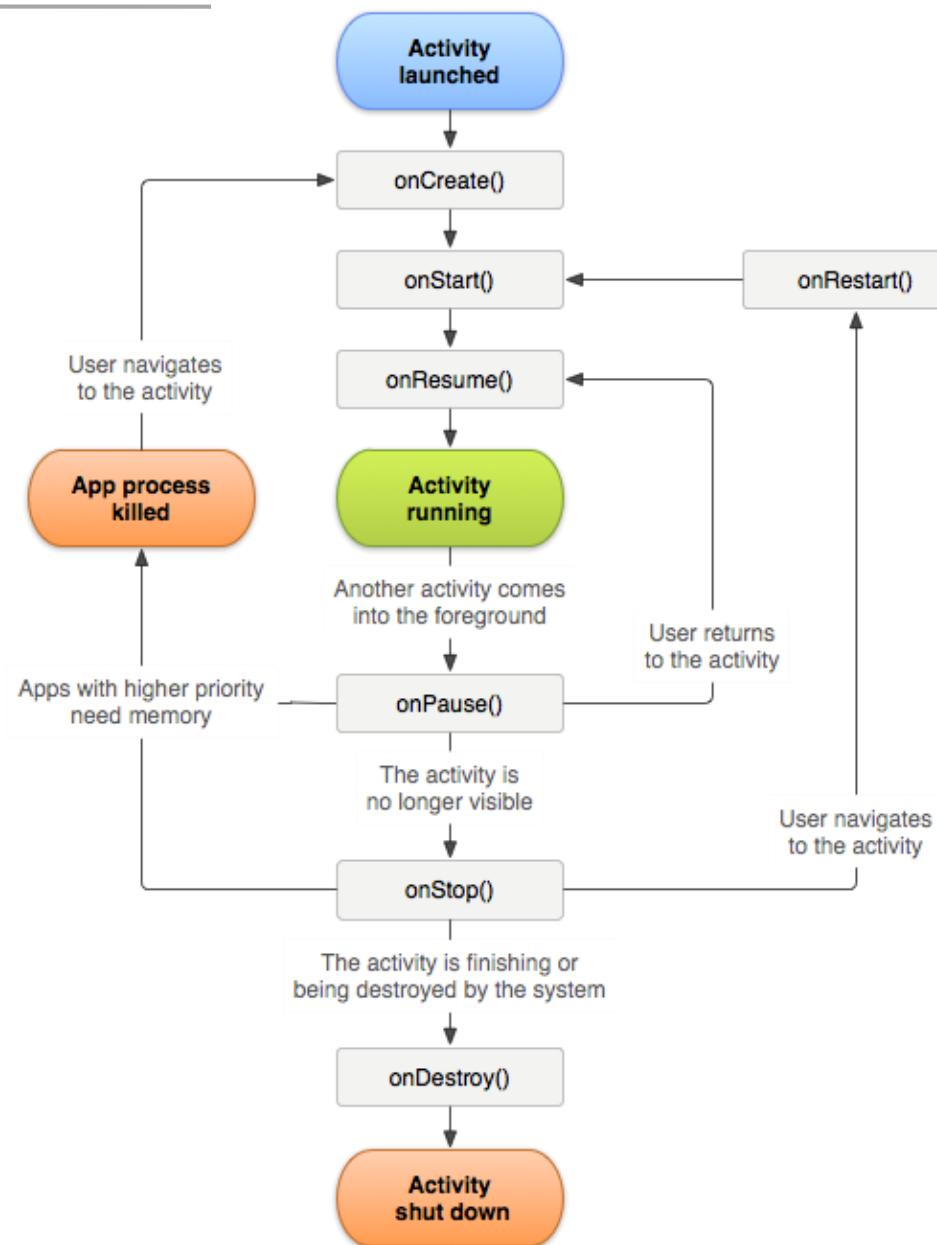


- Bouton abandonné : menu





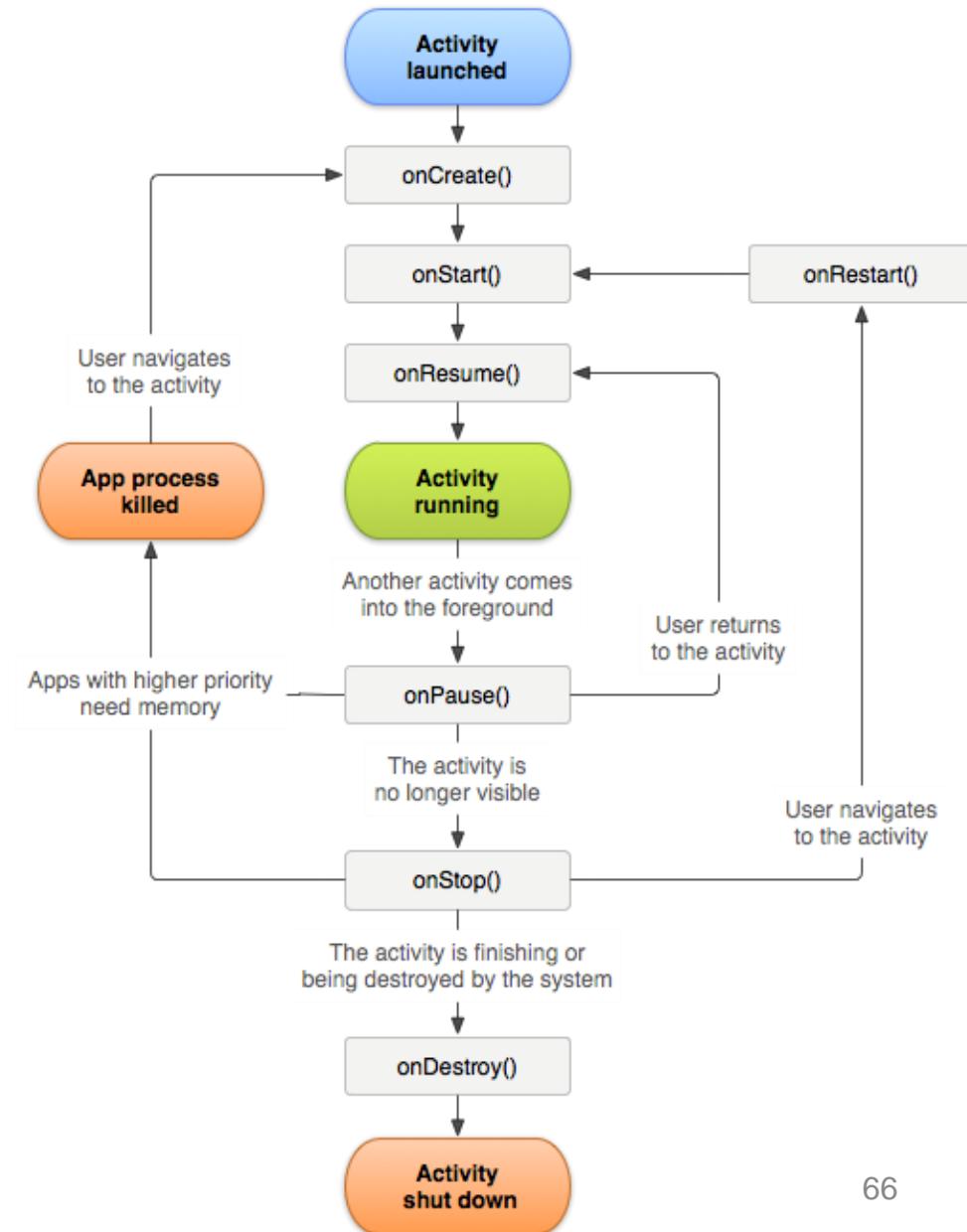
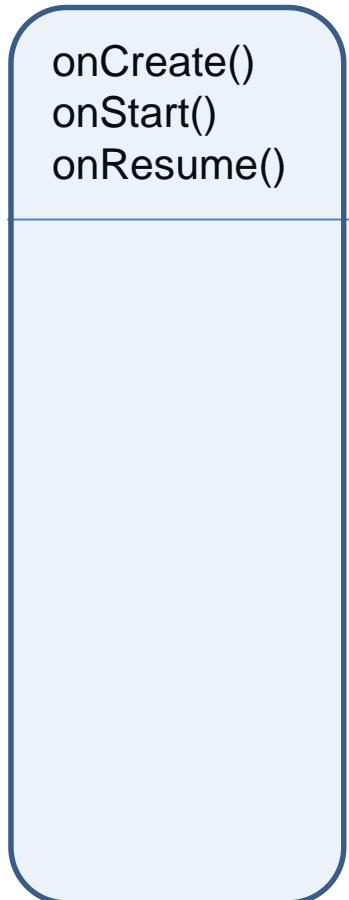
Cycle de vie d'une Activity





Cycle de vie d'une Activity

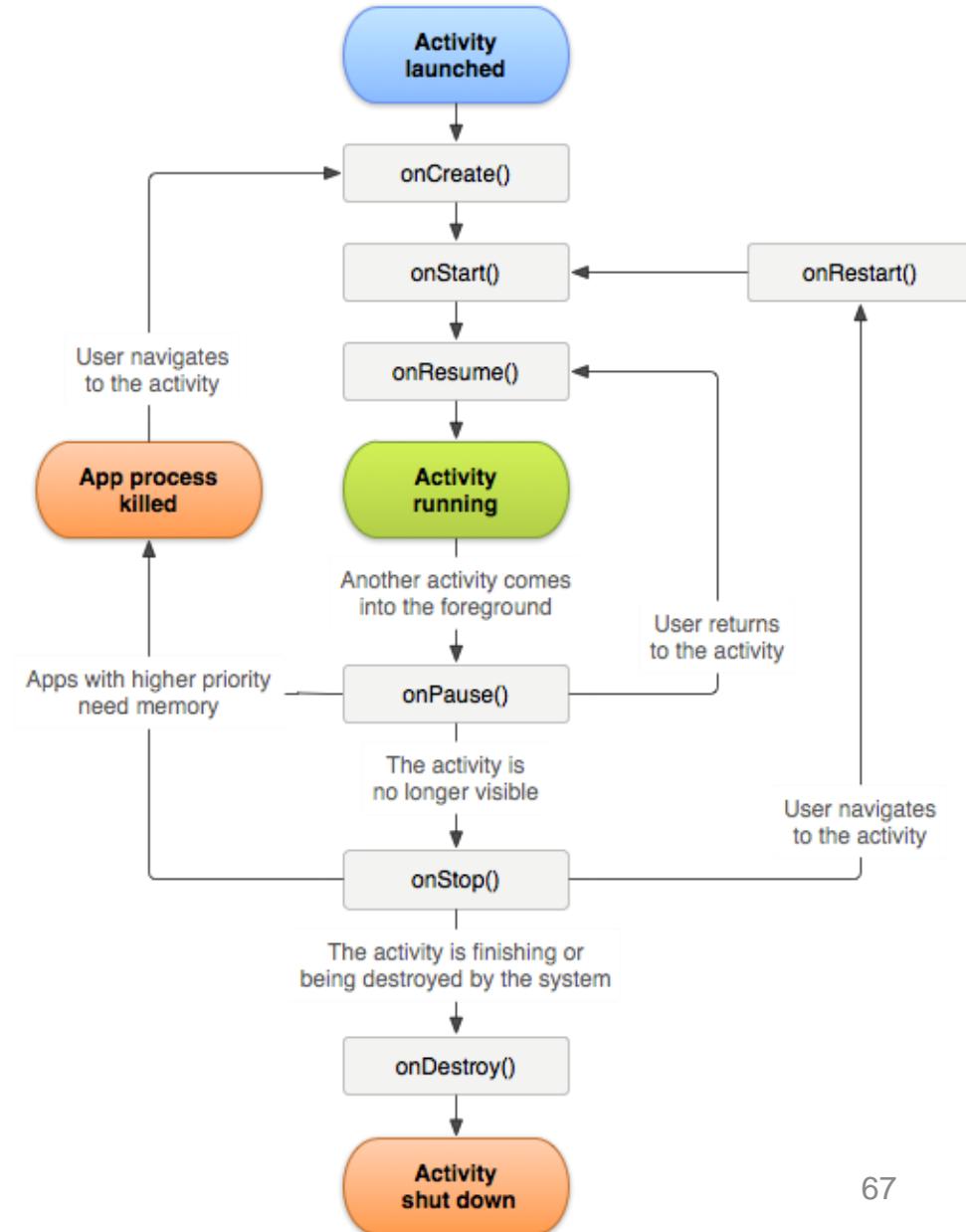
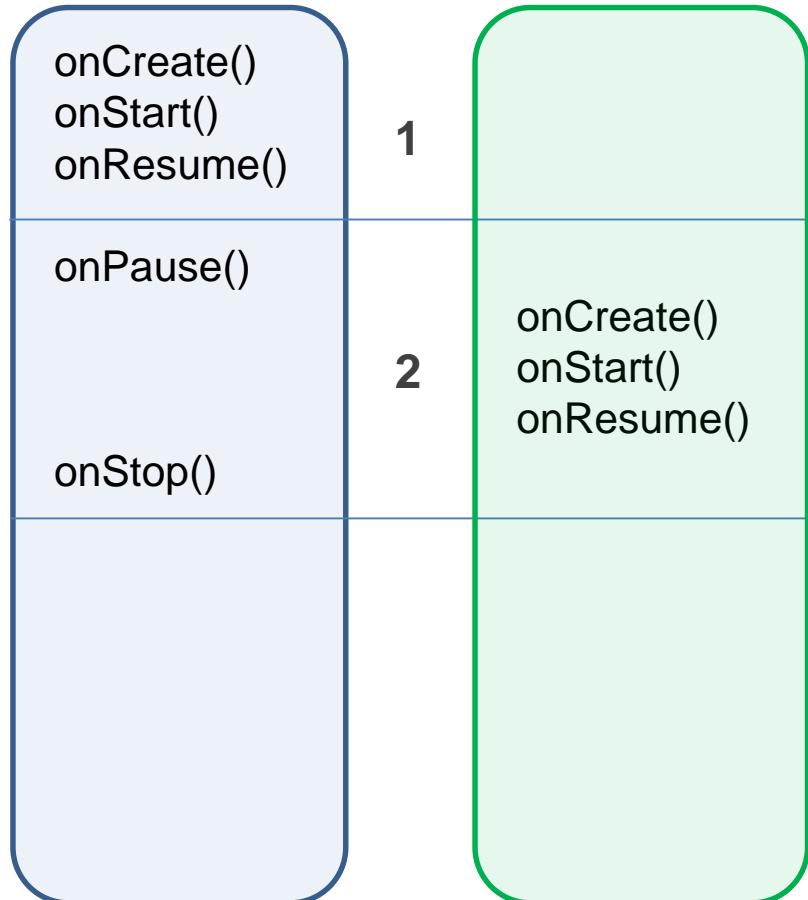
1. Lance l'application (A)





Cycle de vie d'une Activity

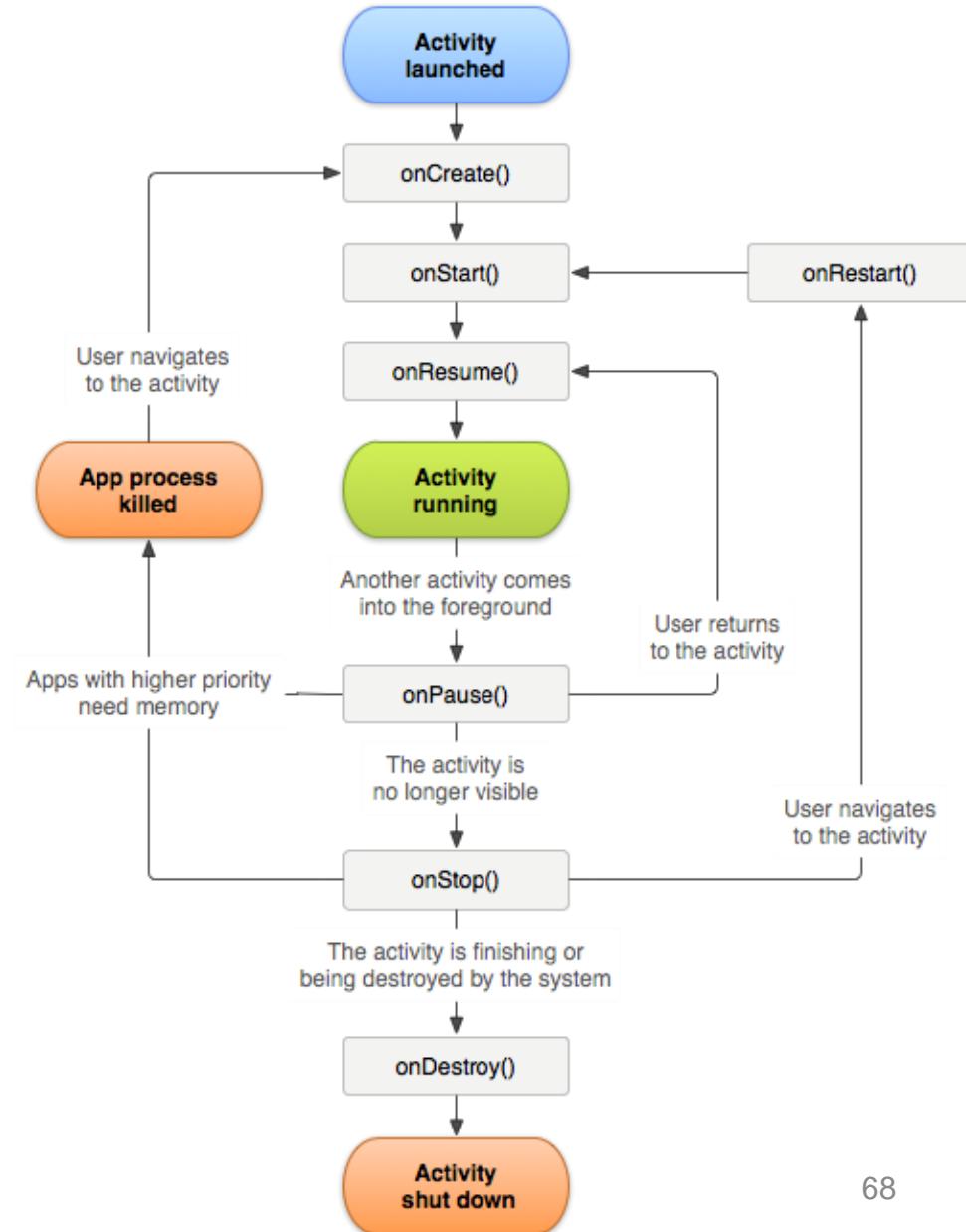
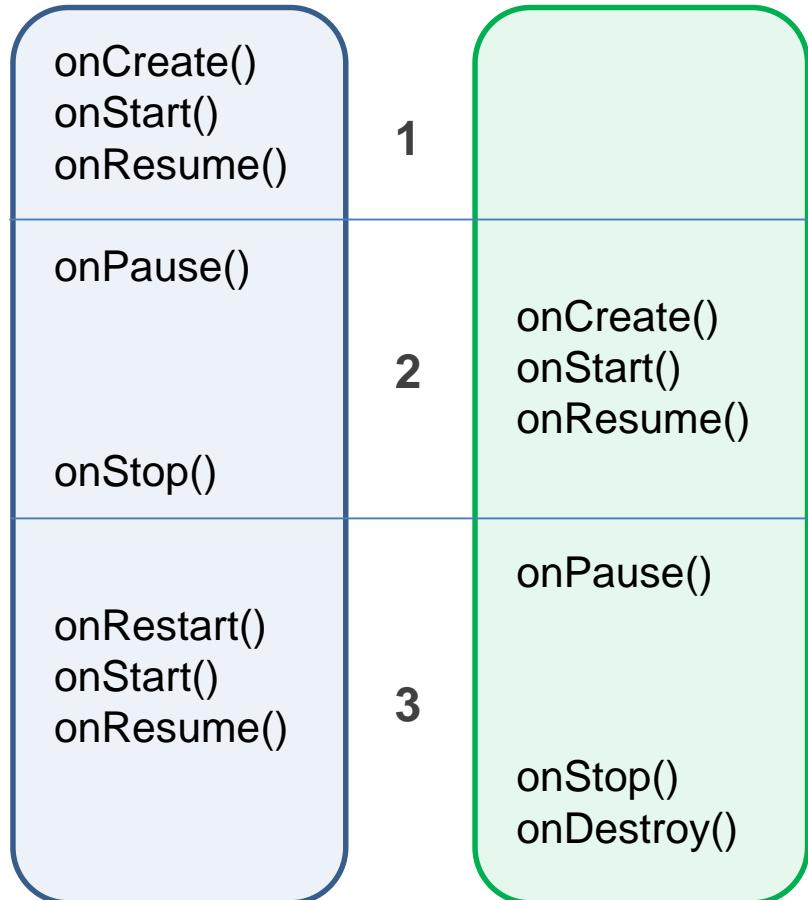
1. Lance l'application (A)
2. L'activity A lance l'activity B





Cycle de vie d'une Activity

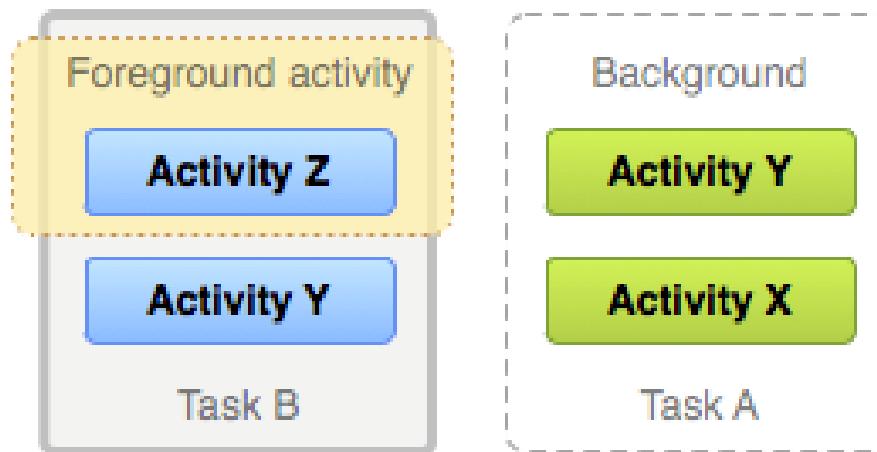
1. Lance l'application (A)
2. L'activity A lance l'activity B
3. Back





Notion de tâches

- Plusieurs tâches sont maintenues par le système
- Une seule contient une Activity visible par l'utilisateur





Manipulation des tâches

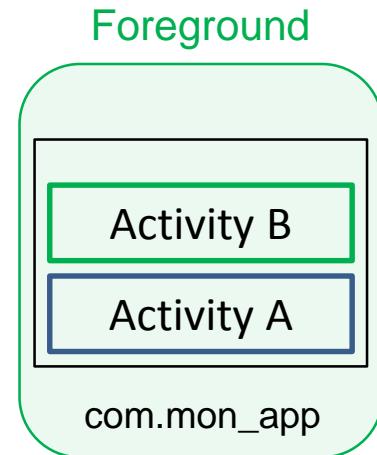
- Mode par défaut assez simple
- Lancement des Activities avec

```
startActivity(new Intent(ActivityX.class));
```

- Exemple 1 : touche back

Scénario

1. Lance l'application (A)
2. A lance B
3. Back





Manipulation des tâches

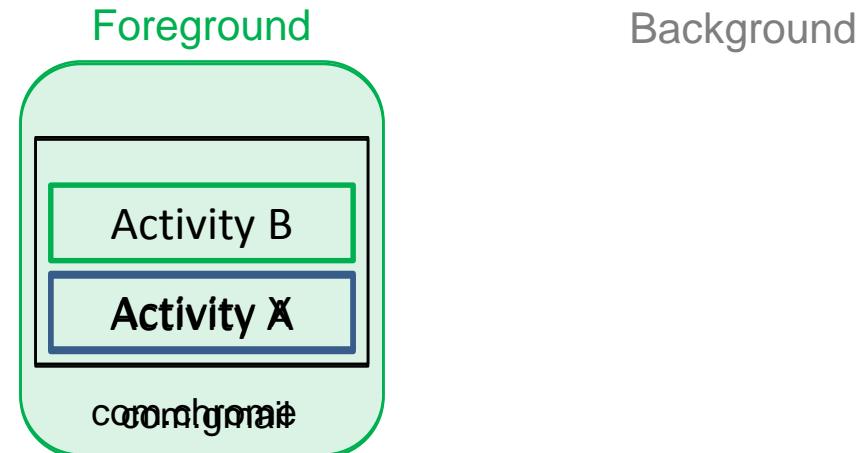
- Mode par défaut assez simple
- Lancement des Activities avec

```
startActivity(new Intent(ActivityX.class));
```

- Exemple 2 : touche home

Scénario

1. Lance l'application Gmail (A)
2. A lance B
3. Home
4. Lance l'application Chrome (X)





Manipulation des tâches

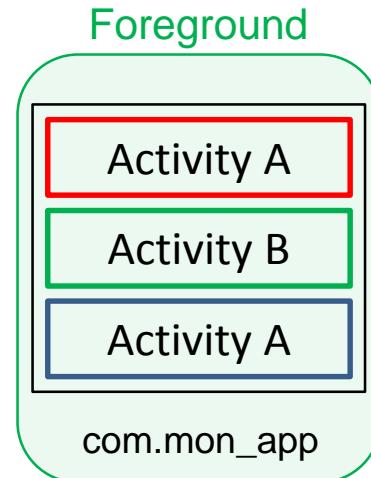
- Mode par défaut assez simple
- Lancement des Activities avec

```
startActivity(new Intent(ActivityX.class));
```

- Exemple 3 : lance 2 fois l'ActivityA

Scénario

1. Lance l'application (A)
2. A lance B
3. B lance A



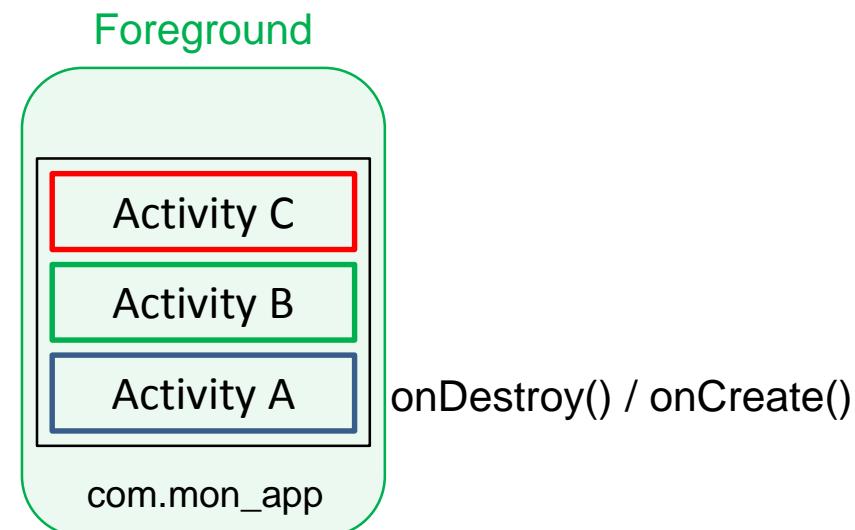


Manipulation des tâches

- Peut être plus complexe
- Flag mode de lancement : `intent.addFlags(Intent.FLAG_**);`
- **Intent.FLAG_ACTIVITY_CLEAR_TOP** : cherche dans la pile si l'Activity existe déjà
 - Activity détruite (`onDestroy()`) puis re-créée (`onCreate()`)

Scénario

1. Lance l'application (A)
2. A lance B
3. B lance C
4. C lance A avec FLAG_ACTIVITY_CLEAR_TOP



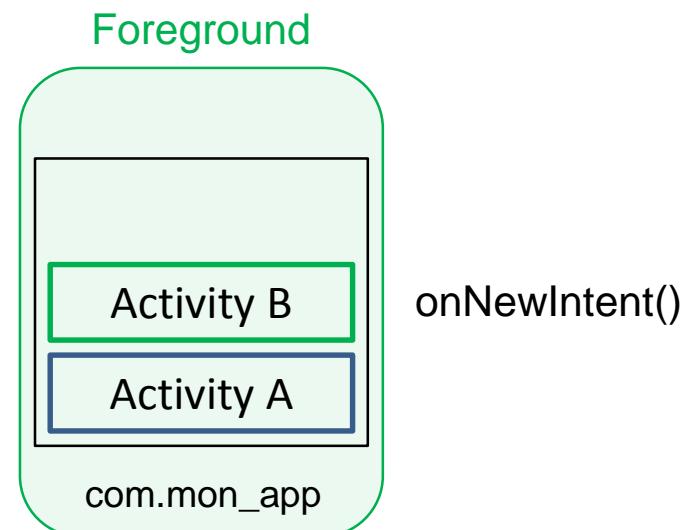


Manipulation des tâches

- Peut être plus complexe
- Flag mode de lancement : `intent.addFlags(Intent.FLAG_**);`
- **Intent.FLAG_ACTIVITY_SINGLE_TOP** : cherche si l'Activity existe déjà en haut de la pile

Scénario

1. Lance l'application (A)
2. A lance B
3. B lance B avec FLAG_ACTIVITY_SINGLE_TOP



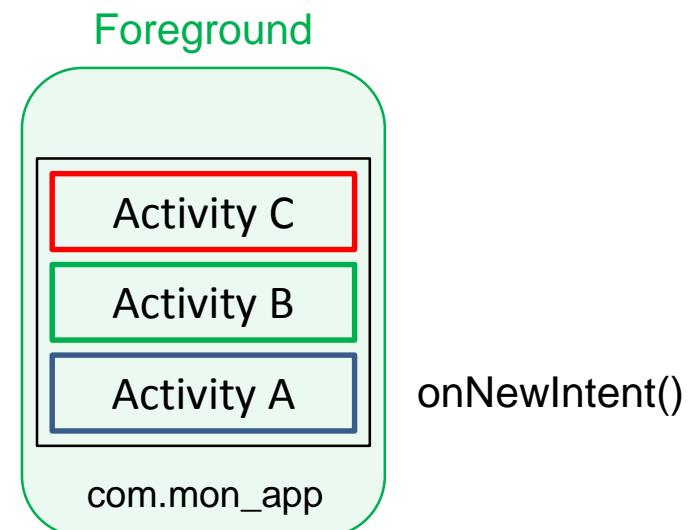


Manipulation des tâches

- Peut être plus complexe
- Flag mode de lancement : `intent.addFlags(Intent.FLAG_**);`
- **Intent.FLAG_ACTIVITY_CLEAR_TOP + Intent.FLAG_ACTIVITY_SINGLE_TOP**

Scénario

1. Lance l'application (A)
2. A lance B
3. B lance C
4. C lance A avec FLAG_ACTIVITY_CLEAR_TOP et FLAG_ACTIVITY_SINGLE_TOP





Manipulation des tâches

- Peut être plus complexe
- Flag mode de lancement : `intent.addFlags(Intent.FLAG_**);`
- Flags :
 - FLAG_ACTIVITY_CLEAR_TOP
 - FLAG_ACTIVITY_SINGLE_TOP
 - FLAG_ACTIVITY_NO_HISTORY
 - FLAG_ACTIVITY_CLEAR_TASK
 - FLAG_ACTIVITY_NEW_TASK
 - ...



Android Studio



Android Studio

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure for "TD_Intro". It includes the "app" module with its "manifests", "java", and "res" directories. The "res" directory contains "drawable", "layout", "mipmap", and "values" folders. The "mipmap" folder specifically contains multiple versions of the "ic_launcher.png" icon.
- Code Editor:** The main area shows the "AndroidManifest.xml" file. The XML code defines the application package "fr.isima.td_intro" and two activities: "ActivityOne" and "ActivityTwo". Both activities are set to be the main launcher activity, indicated by the "MAIN" and "LAUNCHER" intent filters.
- Logcat:** The bottom section shows the log output from the application. It includes logs for both "ActivityOne" and "ActivityTwo", detailing their lifecycle events like onPause, onStart, onResume, onPause, onStop, and onDestroy.
- Bottom Navigation:** The navigation bar at the bottom includes tabs for "Messages", "Terminal", "Android Monitor", "Debug", "TODO", "Event Log", and "Gradle Console".



Android Studio

The screenshot shows the Android Studio interface with the toolbar highlighted by a red box. The toolbar contains various icons for file operations like Open, Save, Cut, Copy, Paste, Find, and Run. Below the toolbar is the main workspace. On the left is the Project Navigational Drawer, which includes sections for Project, Structure, Captures, and Build Variants. The Project section shows the app module with its manifest, Java files (ActivityOne, ActivityTwo, TDActivity), and resources (res folder with drawable, layout, mipmap, and values). The main workspace displays the AndroidManifest.xml file's XML code. At the bottom, there are tabs for Text and Merged Manifest, and a Debug tab showing logcat output. The logcat output shows several log messages from ActivityOne and ActivityTwo. The bottom navigation bar includes tabs for Messages, Terminal, Android Monitor, Debug, TODO, Event Log, and Gradle Console. A status bar at the very bottom provides information about the current session.

TD_Intro - [C:\Users\Roman\Documents\AndroidstudioProjects\TD_Intro] - [app] - ...\\app\\src\\main\\AndroidManifest.xml - Android Studio 2.2.2

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Toolbar

Project

Structure

Captures

Build Variants

main\AndroidManifest.xml

manifest application

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.isima.td_intro" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="TD_Intro"
        android:supportsRtl="true"
        android:theme="@android:style/Theme.Holo.Light.DarkActionBar" >

        <activity android:name=".ActivityOne" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".ActivityTwo"/>
    </application>
</manifest>
```

Text Merged Manifest

Debug app

Debugger Console

I/fr.isima.td_intro.ActivityOne: fr.isima.td.intro.ActivityOne : onPause
I/fr.isima.td_intro.ActivityTwo: fr.isima.td_intro.ActivityTwo : onStart
I/fr.isima.td_intro.ActivityTwo: fr.isima.td_intro.ActivityTwo : onResume
I/art: Do partial code cache collection, code=38KB, data=42KB
I/art: After code cache collection, code=38KB, data=42KB
I/art: Increasing code cache capacity to 256KB
I/art: Compiler allocated 7MB to compile void android.widget.TextView.<init>(android.content.Context, android.util.AttributeSet, int, int)
I/fr.isima.td_intro.ActivityOne: fr.isima.td.intro.ActivityOne : onStop
I/fr.isima.td_intro.ActivityTwo: fr.isima.td_intro.ActivityTwo : onPause
I/fr.isima.td_intro.ActivityOne: fr.isima.td.intro.ActivityOne : onRestart
I/fr.isima.td_intro.ActivityOne: fr.isima.td.intro.ActivityOne : onStart
I/fr.isima.td_intro.ActivityOne: fr.isima.td.intro.ActivityOne : onResume
I/fr.isima.td_intro.ActivityTwo: fr.isima.td.intro.ActivityTwo : onStop
I/fr.isima.td_intro.ActivityTwo: fr.isima.td_intro.ActivityTwo : onDestroy
I/fr.isima.td_intro.ActivityOne: fr.isima.td.intro.ActivityOne : onPause
I/fr.isima.td_intro.ActivityOne: fr.isima.td.intro.ActivityOne : onStop
I/fr.isima.td_intro.ActivityOne: fr.isima.td.intro.ActivityOne : onDestroy

Messages Terminal Android Monitor Debug TODO Event Log Gradle Console

Instant Run detected that you are running on a target device that has a work profile or multiple user accounts. // Launching the a... (5 minutes ago) 5:17 CRLF+ UTF-8 Context: <no context>



Android Studio : toolbar

Configuration du build
(que vous pouvez éditer)

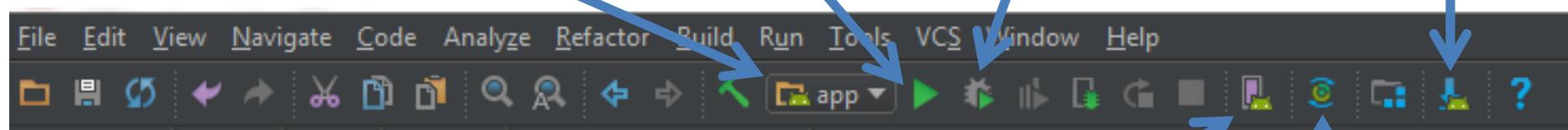
Lancer

Lancer en
mode debug
(Instant Run)

SDK Manager

Gestionnaire
d'émulateurs

Run Gradle
(exécuté automatiquement
lors du lancement)





Android Studio : émulateurs

Virtual Device Configuration X

Select Hardware

Android Studio

Choose a device definition

Category	Name ▾	Size	Resolution	Density
TV	Nexus S	4,0"	480x800	hdpi
Wear	Nexus One	3,7"	480x800	hdpi
Phone	Nexus 6P	5,7"	1440x2560	560dpi
Tablet	Nexus 6	5,96"	1440x2560	560dpi
	Nexus 5X	5,2"	1080x1920	420dpi
	Nexus 5	4,95"	1080x1920	xxhdpi
	Nexus 4	4,7"	768x1280	xhdpi
	Galaxy Nexus	4,65"	720x1280	xhdpi
	5,4" FWVGA	5,4"	480x854	mdpi

Nexus 5

Size: normal
Ratio: long
Density: 420dpi

New Hardware Profile Import Hardware Profiles ⟳ Clone Device... Previous Next Cancel Finish Help



Android Studio : émulateurs

Virtual Device Configuration

Select Hardware

Choose a device definition

Category: Nexus

User: Nexus 5

Nexus One

Nexus 4P

Nexus 5X

Nexus 6

Nexus 6P

Nexus 9

Nexus 10

Nexus 10P

Nexus 4

Samsung Nexus

Samsung Galaxy S6

New Hardware Profile Import Hardware Profiles

Virtual Device Configuration

System Image

Android Studio

Select a system image

Recommended x86 Images Other Images

Release Name	API Level	ABI	Target
Nougat	25	x86	Android 7.1.1 (with Google APIs)
Nougat	25	x86_64	Android 7.1.1 (with Google APIs)
Nougat	24	x86	Android 7.0 (with Google APIs)
Nougat	24	x86_64	Android 7.0 (with Google APIs)
Marshmallow	23	x86	Android 6.0 (with Google APIs)
Marshmallow	23	x86_64	Android 6.0 (with Google APIs)
Lollipop Download	22	x86_64	Android 5.1 (with Google APIs)
Lollipop Download	22	x86	Android 5.1 (with Google APIs)

Marshmallow

API Level 23

Android 6.0

Google Inc.

System Image x86_64

These images are recommended because they run the fastest and include support for Google APIs

Questions on API level?
See the [API level distribution chart](#)

Previous Next Cancel Finish Help



Android Studio

The screenshot shows the Android Studio interface with several key components highlighted:

- Project pane (left):** Shows the project structure for "TD_Intro". It includes the "app" module with its "manifests", "java", and "res" directories. The "res" directory contains "drawable", "layout", "mipmap", and "values" folders. The "mipmap" folder specifically contains multiple versions of "ic_launcher.png" for different screen densities.
- Code editor (top right):** Displays the "AndroidManifest.xml" file. A red box highlights the code area. The XML defines an application with two activities: "ActivityOne" and "ActivityTwo". Both activities are set to be the main launcher activity, indicated by the "MAIN" intent filter and the "LAUNCHER" category.
- Debug / Console pane (bottom right):** Shows the Android Monitor tab, which displays logcat output. A red box highlights this pane. The logcat output shows various system and application logs, including activity lifecycle events like onPause, onStart, onResume, and onDestroy.
- Bottom navigation bar:** Contains tabs for "Messages", "Terminal", "Android Monitor", "Debug", "TODO", "Event Log", and "Gradle Console".

A red box also highlights the word "Code" above the code editor and the word "Debug / Console" above the monitor pane.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.isima.td_intro" >
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="TD_Intro"
        android:supportsRtl="true"
        android:theme="@android:style/Theme.Holo.Light.DarkActionBar" >
        <activity android:name=".ActivityOne" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ActivityTwo"/>
    </application>
</manifest>
```

Instant Run detected that you are running on a target device that has a work profile or multiple user accounts. // Launching the a... (5 minutes ago) 5:17 CRLF+ UTF-8 Context: <no context>



Android Studio : debug

Point d'arrêt
(breakpoint)

Contrôle du mode debug
(avance d'1 instruction,
reprise, stop...)

The screenshot shows the Android Studio interface during a debug session. The code editor displays Java code with several breakpoints set, indicated by red circles with a checkmark. The toolbar below the editor contains various debug control buttons. The bottom navigation bar includes tabs for Messages, Terminal, Android Monitor, Debug (which is selected), and TODO. On the left, there's a sidebar with icons for Favorites, Build Variants, and a question mark. The main window is divided into several sections: a call stack window titled 'Frames' showing the current stack trace, a 'Variables' window listing live variable values with their memory addresses, and a 'Breakpoints' window. The 'Debug' tab is highlighted in the bottom navigation bar.

Point d'arrêt (breakpoint)

Contrôle du mode debug (avance d'1 instruction, reprise, stop...)

Pile d'appels

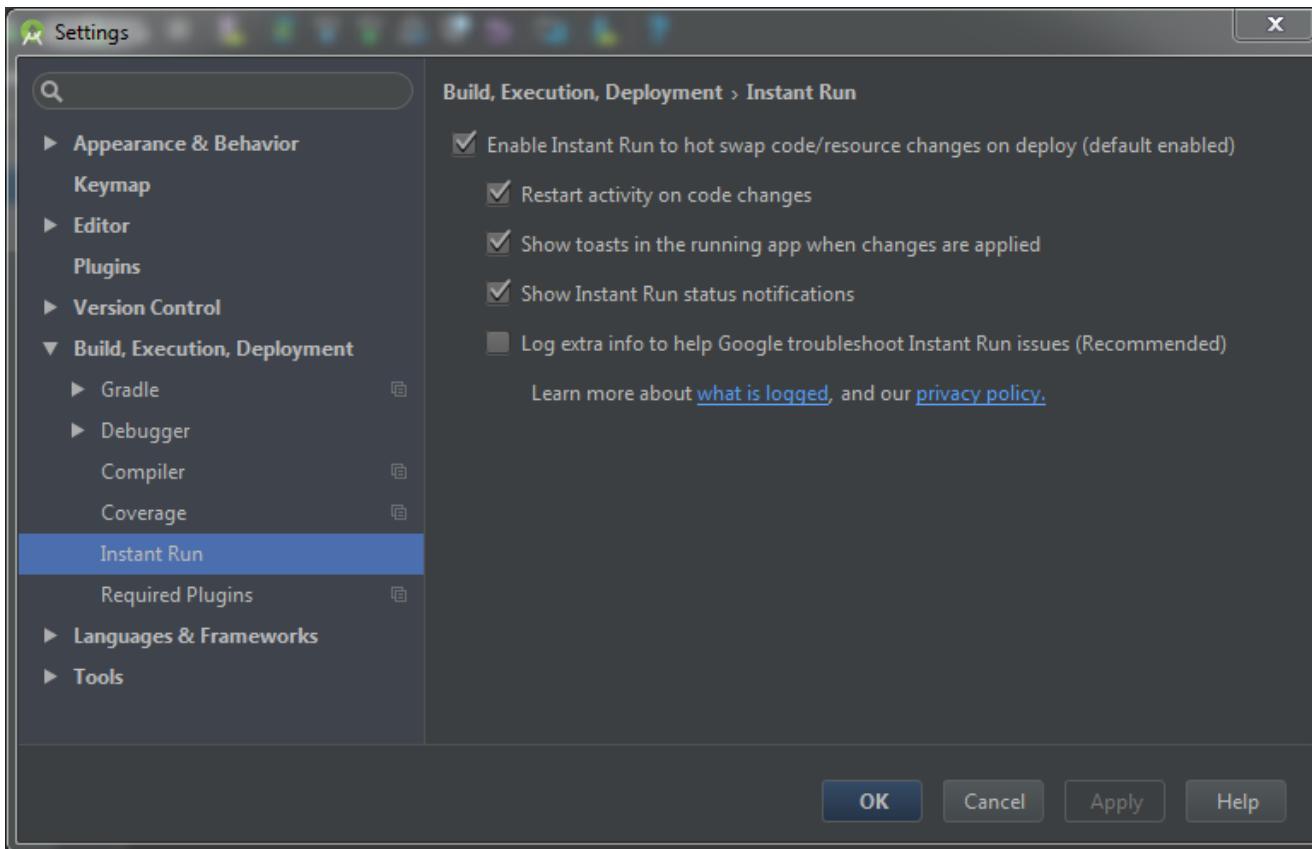
Variables (valeurs en live)



Android Studio : Instant Run

- Vérifiez que l'Instant Run soit bien activé : redéploiement beaucoup plus rapide en mode debug suite à un changement dans le code

File > Settings > Build, Execution, Deployment > Instant Run





Android Studio

The screenshot shows the Android Studio interface with the following components:

- Project Tab:** Displays the project structure with a red border around it. It includes the `app` module with its `manifests`, `java`, and `res` directories. The `res` directory contains `drawable`, `layout`, and `mipmap` folders, with several `ic_launcher.png` files for different densities.
- Editor Tab:** Shows the `main\AndroidManifest.xml` file with the following XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.isima.td_intro" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="TD_Intro"
        android:supportsRtl="true"
        android:theme="@android:style/Theme.Holo.Light.DarkActionBar" >

        <activity android:name=".ActivityOne" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".ActivityTwo"/>
    </application>
</manifest>
```
- Logcat Tab:** Displays log messages from the application's runtime. Key messages include:

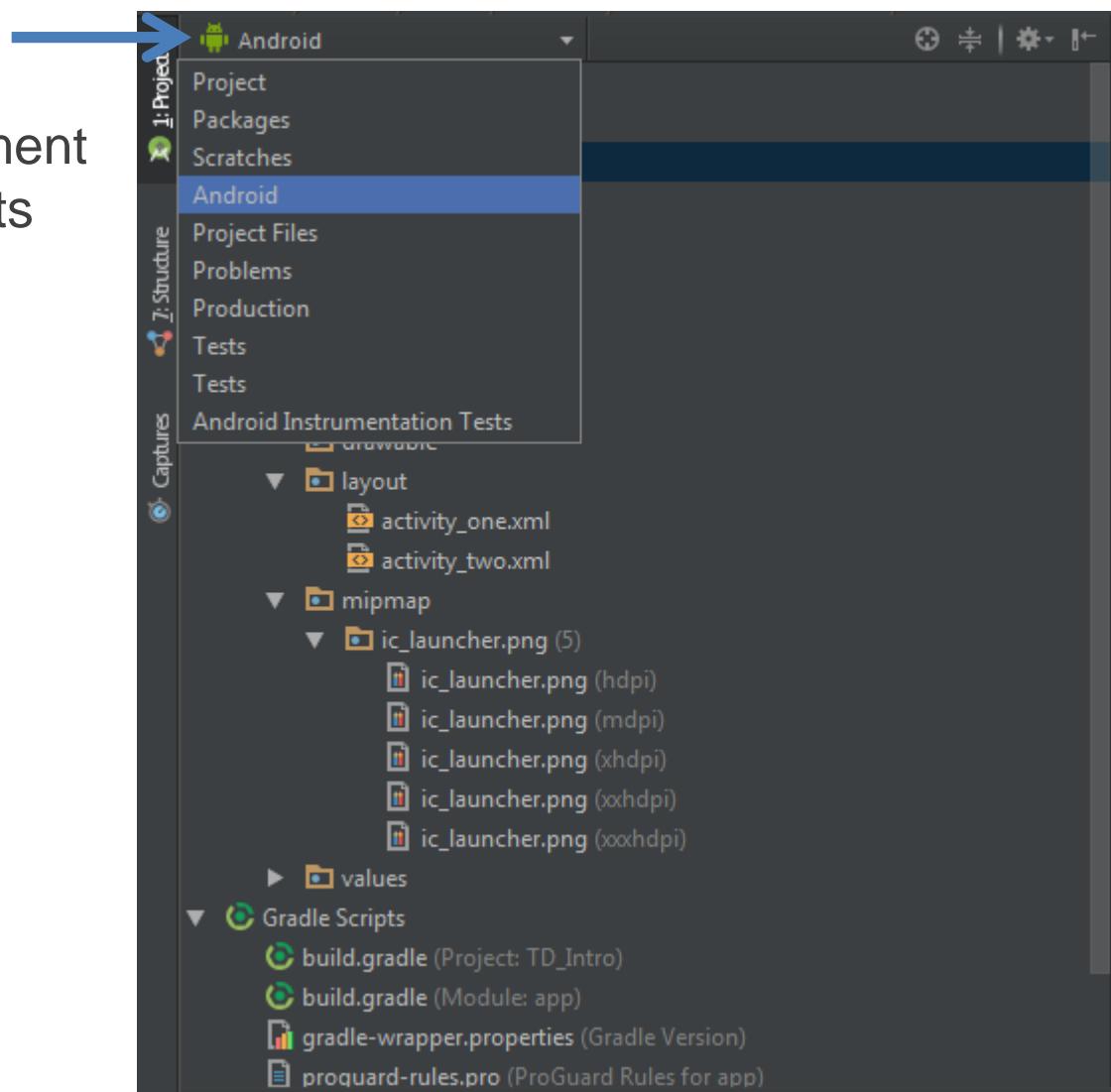
```
I/fr.isima.td_intro.ActivityOne: fr.isima.td_intro.ActivityOne : onPause
I/fr.isima.td_intro.ActivityTwo: fr.isima.td_intro.ActivityTwo : onStart
I/fr.isima.td_intro.ActivityTwo: fr.isima.td_intro.ActivityTwo : onResume
I/art: Do partial code cache collection, code=38KB, data=42KB
I/art: After code cache collection, code=38KB, data=42KB
I/art: Increasing code cache capacity to 256KB
I/art: Compiler allocated 7MB to compile void android.widget.TextView.<init>(android.content.Context, android.util.AttributeSet, int, int)
I/fr.isima.td_intro.ActivityOne: fr.isima.td_intro.ActivityOne : onStop
I/fr.isima.td_intro.ActivityTwo: fr.isima.td_intro.ActivityTwo : onPause
I/fr.isima.td_intro.ActivityOne: fr.isima.td_intro.ActivityOne : onRestart
I/fr.isima.td_intro.ActivityOne: fr.isima.td_intro.ActivityOne : onStart
I/fr.isima.td_intro.ActivityOne: fr.isima.td_intro.ActivityOne : onResume
I/fr.isima.td_intro.ActivityTwo: fr.isima.td_intro.ActivityTwo : onStop
I/fr.isima.td_intro.ActivityTwo: fr.isima.td_intro.ActivityTwo : onDestroy
I/fr.isima.td_intro.ActivityOne: fr.isima.td_intro.ActivityOne : onPause
I/fr.isima.td_intro.ActivityOne: fr.isima.td_intro.ActivityOne : onStop
I/fr.isima.td_intro.ActivityOne: fr.isima.td_intro.ActivityOne : onDestroy
```



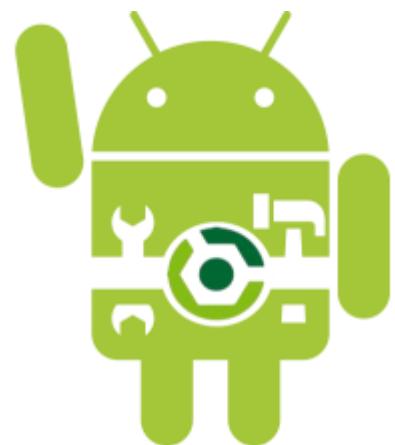
Android Studio : vue projet

- Types de vue :

- Projet Android (regroupement des ressources, des scripts gradle...)
- Projet classique
- Tests
- ...

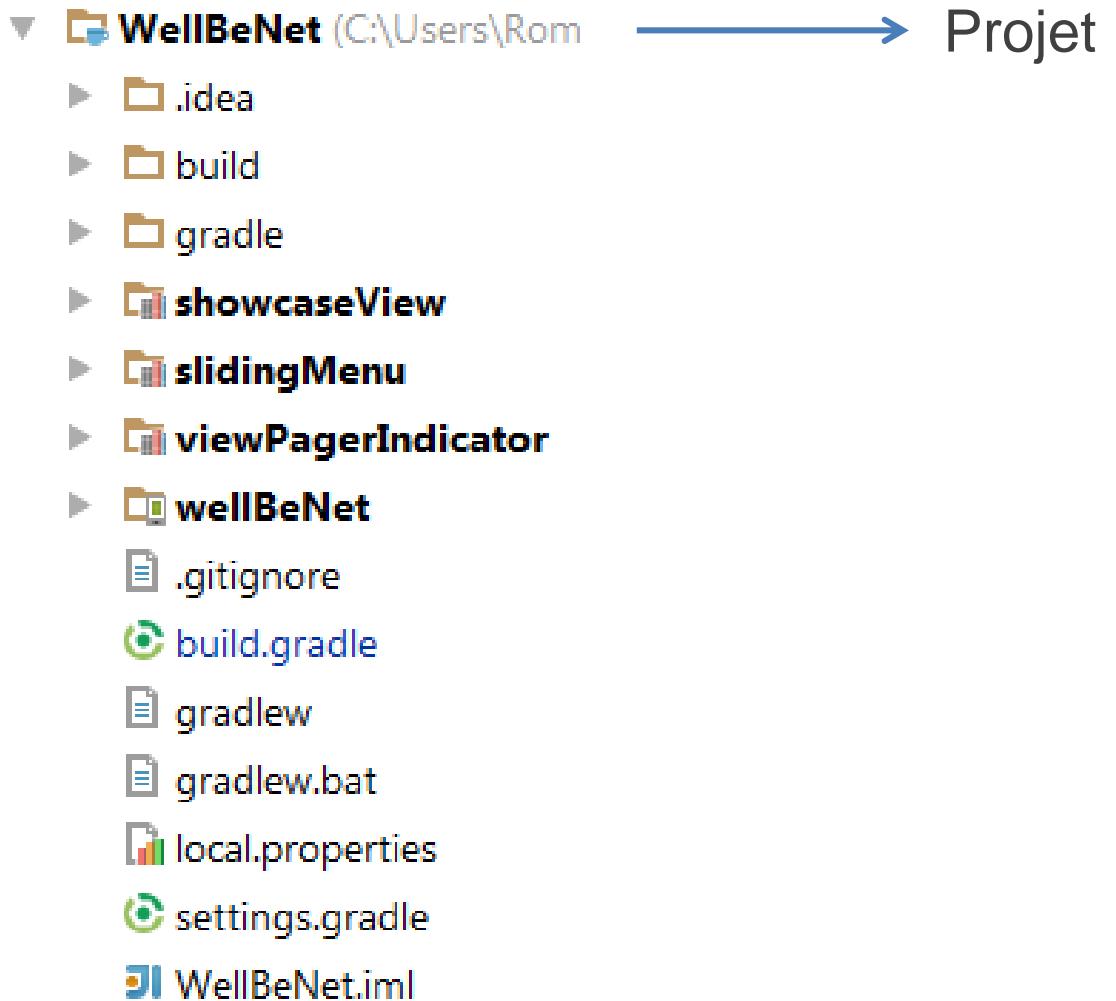


Structure d'un projet et présentation de Gradle



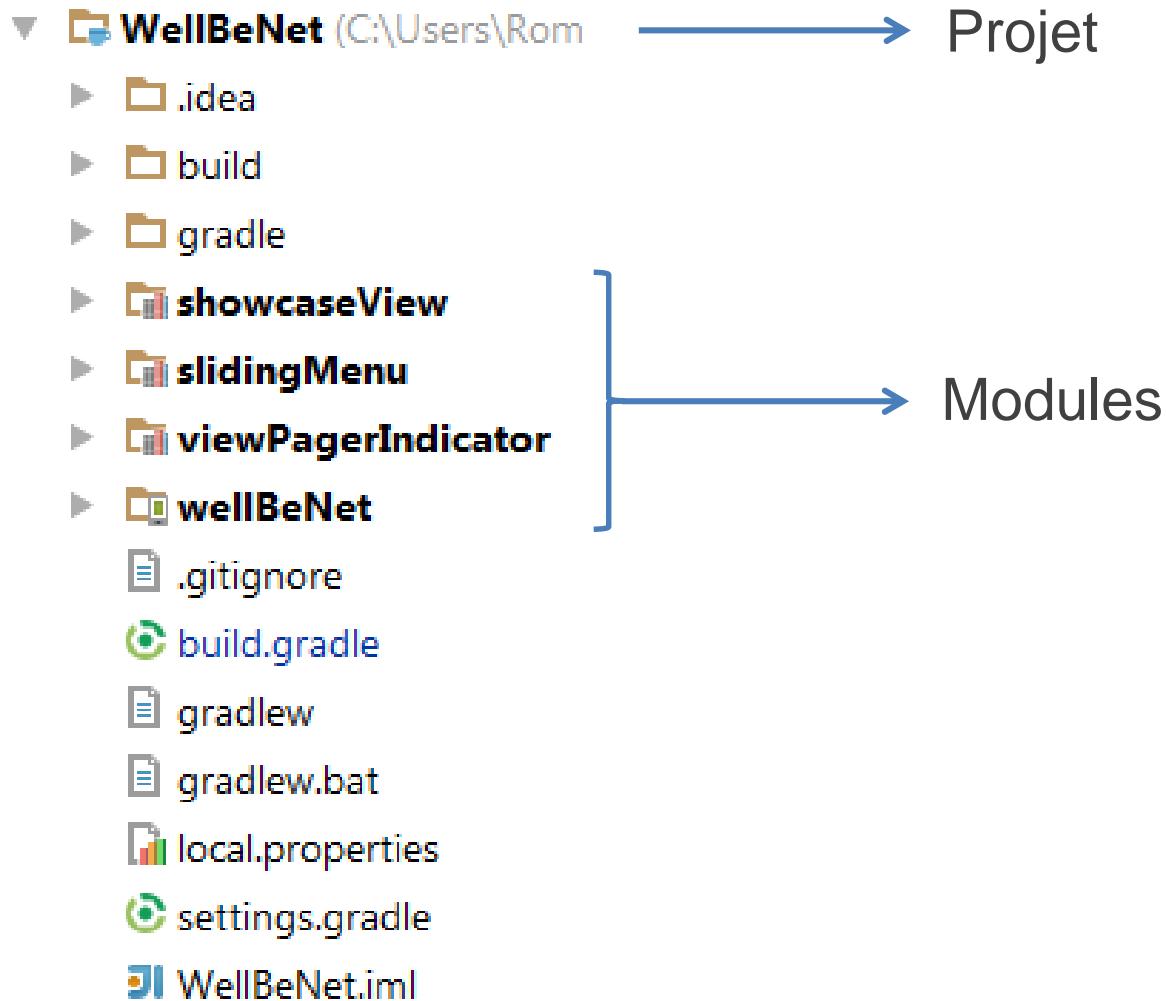


Structure d'un projet : vue générale



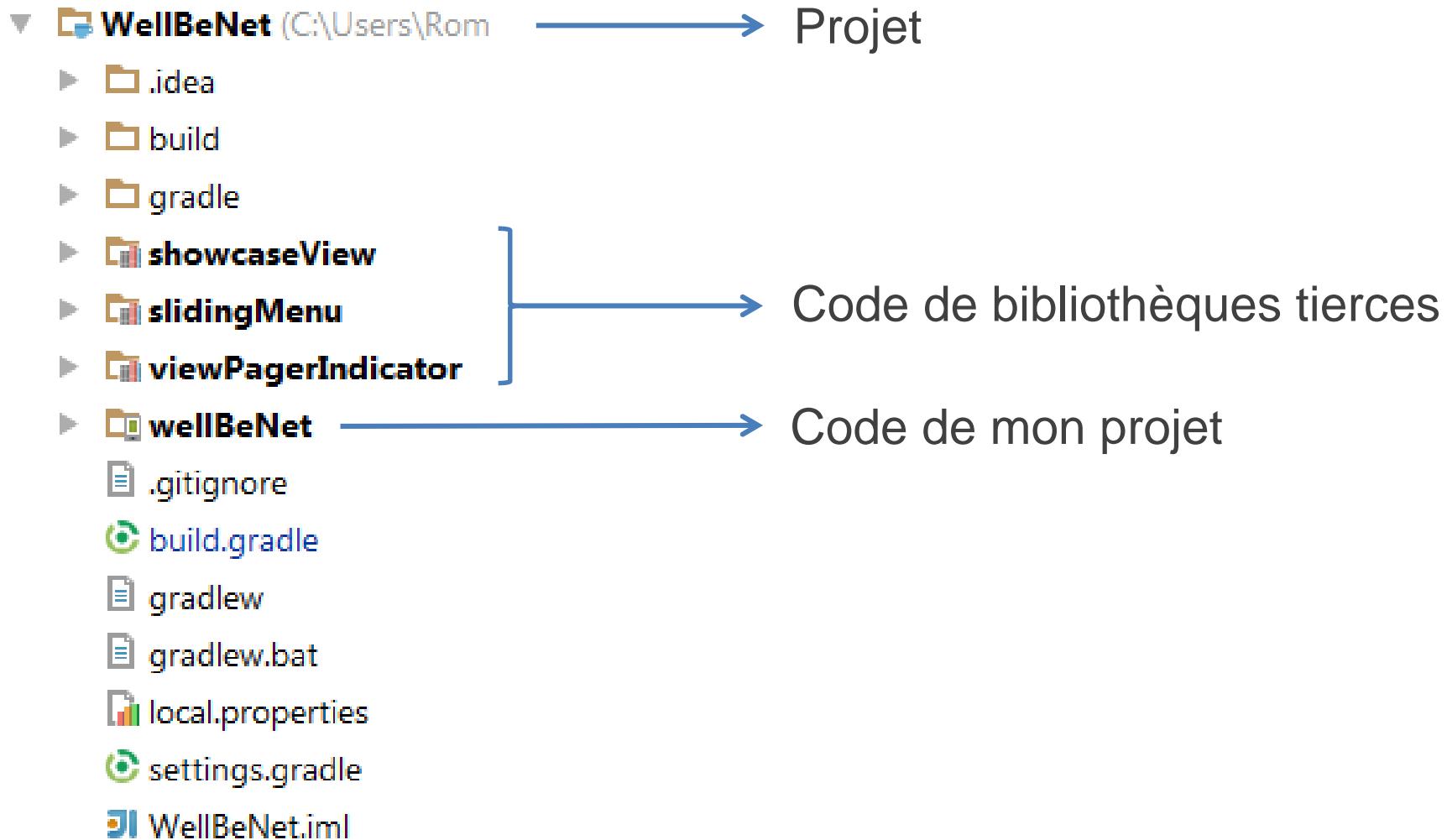


Structure d'un projet : vue générale



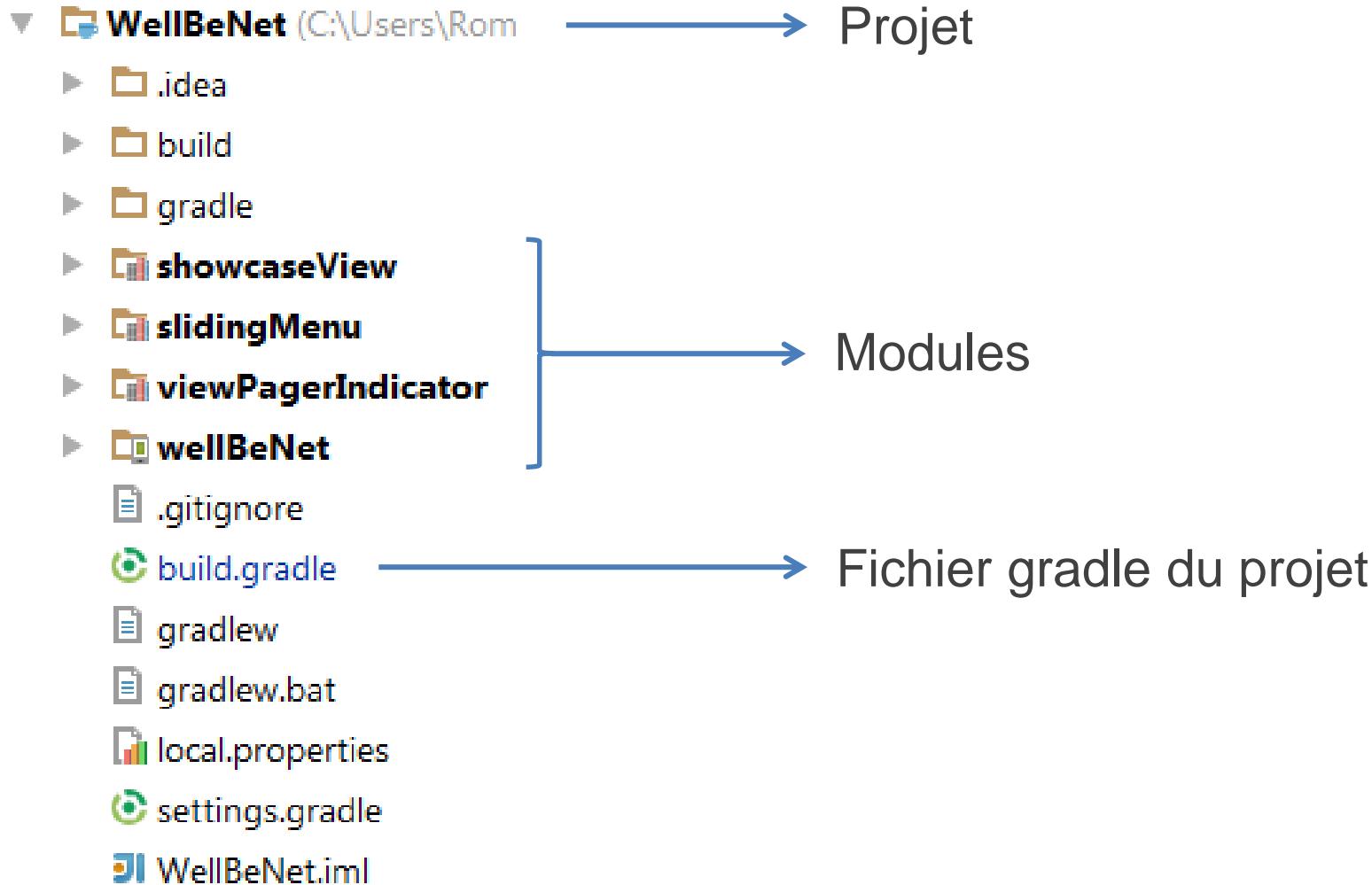


Structure d'un projet : vue générale



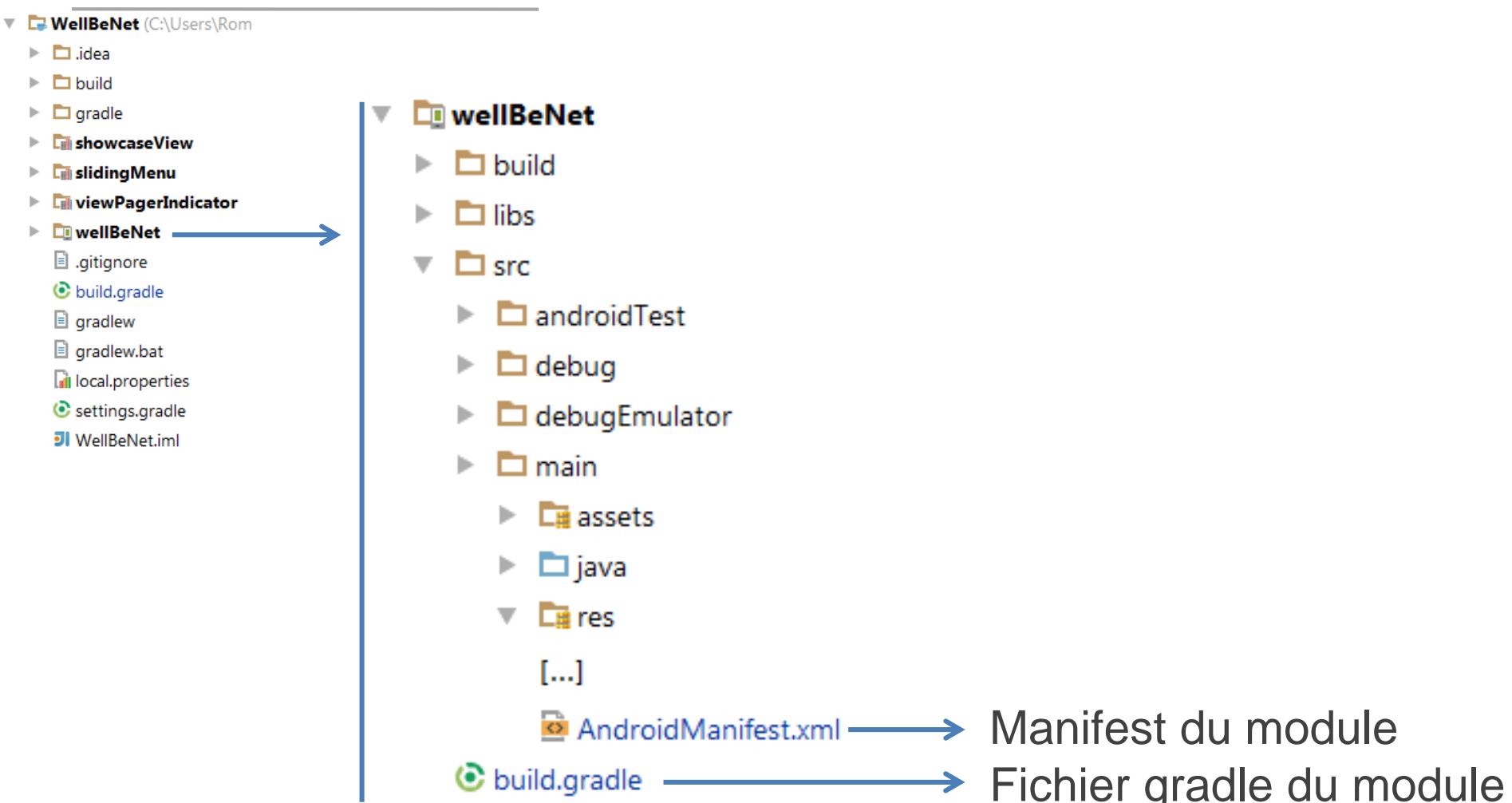


Structure d'un projet : vue générale





Zoom sur un module





Qu'est-ce que Gradle ?

- Outil de gestion et d'automatisation de production de projets, comme Ant ou Maven
- C'est un Domain-Specific Language (DSL) basé sur Groovy (Groovy : langage orienté objet qui tourne sur une JVM)
 - Maven et Ant : fichiers XML
- Google a créé un plugin : Gradle for Android
 - Permet de builder des projets Android
- Android Studio est fourni avec Gradle et Gradle for Android préinstallés



Configuration de Gradle

- 1 fichier gradle à la racine du projet
- Indique les dépendances nécessaires aux scripts de build (configuration de Gradle)

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:2.2.2'  
    }  
}  
  
allprojects {  
    repositories {  
        jcenter()  
    }  
}
```



Configuration de Gradle

- 1 fichier gradle à la racine du projet
- Indique les dépendances nécessaires aux scripts de build (configuration de Gradle)

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:2.2.2'  
    }  
}  
  
allprojects {  
    repositories {  
        jcenter()  
    }  
}
```

D'où peuvent venir les dépendances ?
(Ici jcenter, où Google publie ses bibliothèques)



Configuration de Gradle

- 1 fichier gradle à la racine du projet
- Indique les dépendances nécessaires aux scripts de build (configuration de Gradle)

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:2.2.2'  
    }  
}  
allprojects {  
    repositories {  
        jcenter()  
    }  
}
```

Quelles sont
les dépendances ?
(Ici Gradle for Android)



Configuration d'un module

- 1 fichier gradle par module

```
apply plugin: 'com.android.application'
```

```
android {  
    compileSdkVersion 23  
    buildToolsVersion "23.0.1"
```

```
[...]  
}
```

```
dependencies {  
    compile 'com.android.support:support-v4:23.1.0'  
}
```



Configuration d'un module

- 1 fichier gradle par module

```
apply plugin: 'com.android.application'
```

On construit une application Android
(avec le plugin Gradle for Android)

```
android {  
    compileSdkVersion 23  
    buildToolsVersion "23.0.1"
```

```
[...]  
}
```

```
dependencies {  
    compile 'com.android.support:support-v4:23.1.0'  
}
```



Configuration d'un module

- 1 fichier gradle par module

```
apply plugin: 'com.android.application'
```

```
android {  
    compileSdkVersion 23  
    buildToolsVersion "23.0.1"  
  
    [...]  
}
```

```
dependencies {  
    compile 'com.android.support:support-v4:23.1.0'  
}
```

Quelles sont les dépendances du module?
(Ici l'Android Support Library)



Configuration d'un module

- 1 fichier gradle par module

```
apply plugin: 'com.android.application'
```

```
android {  
    compileSdkVersion 23  
    buildToolsVersion "23.0.1"  
  
    [...]  
}
```

```
dependencies {  
    compile 'com.android.support:support-v4:23.1.0'  
}
```

Configuration spécifique à Android



Options de compilation

```
apply plugin: 'com.android.application'
```

```
android {  
    compileSdkVersion 23 —————> Compile avec Android Marshmallow  
    buildToolsVersion "23.0.1"
```

```
[...]  
}
```

```
dependencies {  
    compile 'com.android.support:support-v4:23.1.0'  
}
```



Options de compilation

```
apply plugin: 'com.android.application'
```

```
android {  
    compileSdkVersion 'Google Inc.:Google APIs:23'  
    buildToolsVersion "23.0.1"
```



Compile avec
Android Marshmallow
+ Google APIs

```
[...]  
}
```

```
dependencies {  
    compile 'com.android.support:support-v4:23.1.0'  
}
```



Le Manifest

- Fichier XML localisé dans le répertoire src/main d'un module
- Contient le nom du package de l'application => application ID
 - 2 applis ayant le même app ID ne peuvent pas être installées simultanément sur le téléphone.
 - Le Play Store n'autorise pas 2 applis à avoir le même app ID.
- On y déclare aussi :
 - Les éléments de l'application : Activities, services...
 - La version d'Android minimale.
 - Les autorisations demandées par l'application (accès à Internet, au micro, au GPS...).
 - Le matériel requis (accéléromètre, gyroscope...).
 - ...



Exemple de Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.isima.android.tp1"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/appName"
        android:theme="@style/AppTheme">

        <activity android:name=".SortedLocationsListActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

        <activity android:name=".LocationsListActivity"/>
        <activity android:name=".EditLocationActivity"/>
    </application>
</manifest>
```



Exemple de Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.isima.android.tp1"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/appName"
        android:theme="@style/AppTheme">

        <activity android:name=".SortedLocationsListActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

        <activity android:name=".LocationsListActivity"/>
        <activity android:name=".EditLocationActivity"/>
    </application>
</manifest>
```

Package de l'application



Exemple de Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="fr.isima.android.tp1"  
    android:versionCode="1"  
    android:versionName="1.0">
```

```
<uses-sdk  
    android:minSdkVersion="14"  
    android:targetSdkVersion="19"/>
```

Nécessite une version d'Android
≥ Ice Cream Sandwich

```
<application  
    android:allowBackup="true"  
    android:icon="@drawable/ic_launcher"  
    android:label="@string/appName"  
    android:theme="@style/AppTheme">
```

```
<activity android:name=".SortedLocationsListActivity">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN"/>  
        <category android:name="android.intent.category.LAUNCHER"/>  
    </intent-filter>  
</activity>
```

```
<activity android:name=".LocationsListActivity"/>  
    <activity android:name=".EditLocationActivity"/>
```

```
</application>
```

```
</manifest>
```



Exemple de Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.isima.android.tp1"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/appName"
        android:theme="@style/AppTheme">

        <activity android:name=".SortedLocationsListActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

        <activity android:name=".LocationsListActivity"/>
        <activity android:name=".EditLocationActivity"/>
    </application>
</manifest>
```

Activity principale
(visible dans le Launcher)



Exemple de Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.isima.android.tp1"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/appName"
        android:theme="@style/AppTheme">

        <activity android:name=".SortedLocationsListActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

        <activity android:name=".LocationsListActivity"/>
        <activity android:name=".EditLocationActivity"/>
    </application>
</manifest>
```

Autres Activities



Gradle et le Manifest

- Quelques éléments du Manifest peuvent aussi être définis dans le build.gradle du module
 - Nom du package
 - Application ID (pour mettre autre chose que le nom du package)
 - Version minimale
 - ...
- La config Gradle l'emporte sur la config du Manifest



Gradle et le Manifest

- Exemple de configuration via Gradle :

```
android {  
    compileSdkVersion 23  
    buildToolsVersion "23.0.1"
```

```
    defaultConfig {  
        applicationId "fr.isima.android.tp1"  
        minSdkVersion 14  
        targetSdkVersion 19  
    }  
}
```

Configuration
(écrase le minSdk et targetSdk du Manifest)



Retour sur la structure d'un module

wellBeNet

- ▶ build
- ▶ libs
- ▶ src
 - ▶ main
 - ▶ assets
 - ▶ java
 - ▶ res
 - ▶ animator
 - ▶ color
 - ▶ drawable
 - ▶ drawable-hdpi
 - ▶ drawable-ldpi
 - ▶ drawable-mdpi
 - ▶ drawable-xhdpi
 - ▶ layout
 - ▶ menu
 - ▶ values
 - ▶ values-en
 - ▶ values-v21
 - ▶ xml
 - ▶ AndroidManifest.xml

Code source Java
(Activities, services..., classes...)



Retour sur la structure d'un module

wellBeNet

- ▶ build
- ▶ libs
- ▶ src
 - ▶ main
 - ▶ assets
 - ▶ java
 - ▶ res
 - ▶ animator
 - ▶ color
 - ▶ drawable
 - ▶ drawable-hdpi
 - ▶ drawable-ldpi
 - ▶ drawable-mdpi
 - ▶ drawable-xhdpi
 - ▶ layout
 - ▶ menu
 - ▶ values
 - ▶ values-en
 - ▶ values-v21
 - ▶ xml
 - ▶ AndroidManifest.xml

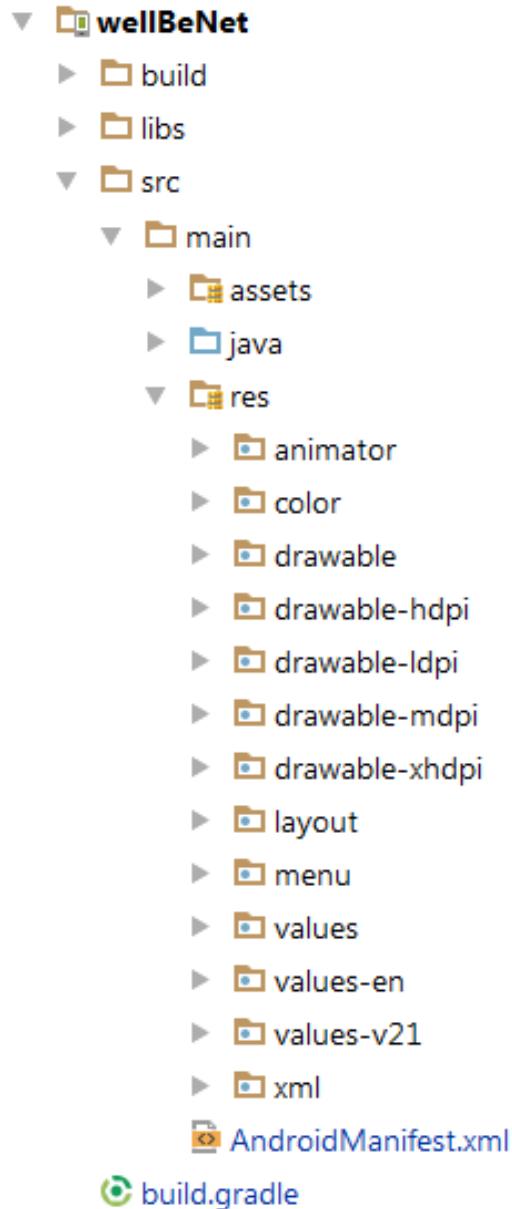
→ Bibliothèques tierces

libs

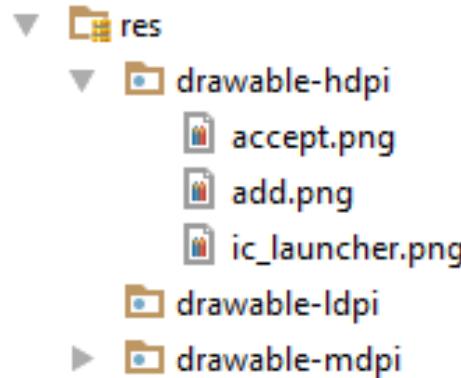
- ▶ jackson-annotations-2.0.5.jar
- ▶ jackson-core-2.0.5.jar
- ▶ jackson-databind-2.0.5.jar
- ▶ jama-1.0.2.jar



Retour sur la structure d'un module

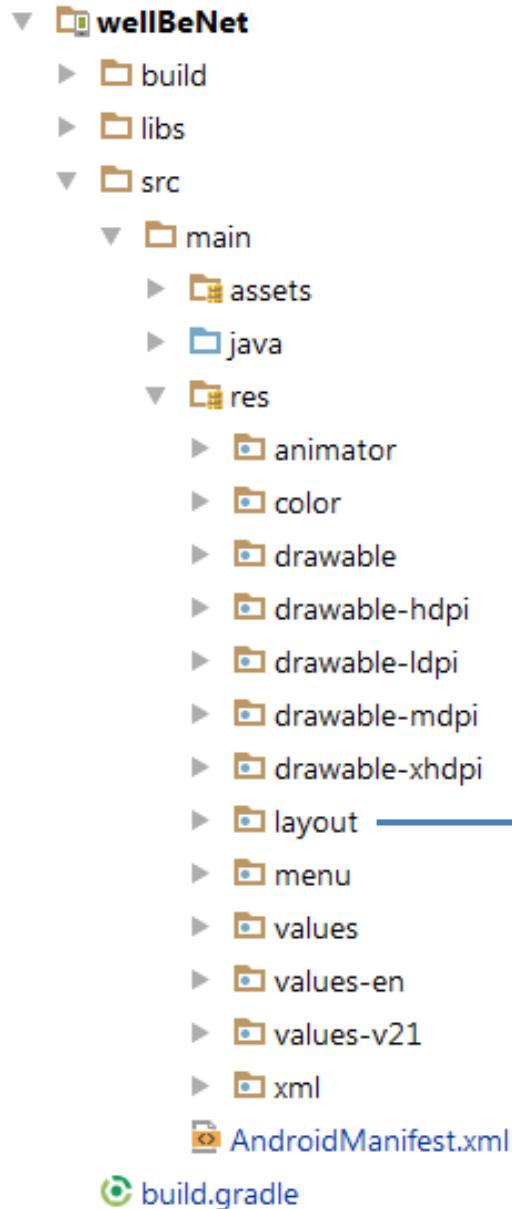


Icônes, images, etc





Retour sur la structure d'un module



Définition des interfaces graphiques
(fichiers XML)



Retour sur la structure d'un module

wellBeNet

- ▶ build
- ▶ libs
- ▼ src
 - ▶ main
 - ▶ assets
 - ▶ java
 - ▼ res
 - ▶ animator
 - ▶ color
 - ▶ drawable
 - ▶ drawable-hdpi
 - ▶ drawable-ldpi
 - ▶ drawable-mdpi
 - ▶ drawable-xhdpi
 - ▶ layout
 - ▶ menu
 - ▶ values
 - ▶ values-en
 - ▶ values-v21
 - ▶ xml
 - AndroidManifest.xml

```
<resources>

    <string name="appName">TP1</string>
    <string name="menuAddLocation">Nouveau lieu</string>
    <string name="LocationName">Nom</string>
    <string name="locationAddress">Adresse</string>
    <string name="menuSaveLocation">Sauvegarder</string>
    <string name="myFavorites">Mes lieux favoris</string>
    <string name="editLocation">Nouveau lieu</string>

</resources>
```

{}

Chaînes de caractères (i18n)
Styles...



Accès aux ressources en Java

- Classe R
 - Générée automatiquement
 - Fourni des identifiants vers les ressources

```
<resources>

    <string name="appName">TP1</string>
    <string name="menuAddLocation">Nouveau lieu</string>
    <string name="LocationName">Nom</string>
    <string name="LocationAddress">Adresse</string>
    <string name="menuSaveLocation">Sauvegarder</string>
    <string name="myFavorites">Mes lieux favoris</string>
    <string name="editLocation">Nouveau lieu</string>

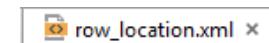
</resources>
```

R.layout.row_location
R.id.name
R.id.address

findViewById(widgetId)

R.string.appName
R.string.menuAddLocation

getText(stringId)



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <TextView
        android:id="@+id/address"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

</LinearLayout>
```



Accès aux ressources en Java

- Principales ressources :

R.string	Chaînes de caractères	res/values <string>
R.layout	Fichiers de layout (UI)	res/layout
R.id	Widgets (TextView...)	widgets dans un layout
R.raw	Ressources brutes	res/raw
R.drawable	Images (ou formes xml)	res/drawable(-*)

- Autres ressources possibles :

R.xml	R.menu
R.anim	R.dimen
R.color	R.array
R.style	R.plurals
R.transition	...



Accès aux ressources du SDK

- Android fournit un ensemble de ressources (layouts, couleurs...)

android.R.layout.simple_list_item_1

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2006 The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-->

<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/text1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceListItemSmall"
    android:gravity="center_vertical"
    android:paddingStart="?android:attr/listPreferredItemPaddingStart"
    android:paddingEnd="?android:attr/listPreferredItemPaddingEnd"
    android:minHeight="?android:attr/listPreferredItemHeightSmall" />
```

TD1 : Visualisation du cycle de vie

Base du TD disponible sur
<https://goo.gl/n3xC3U>

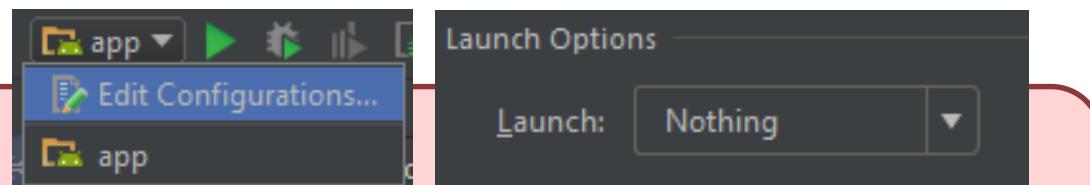




Première application

- Naviguez entre les Activities
- Analysez leur cycle de vie
- Utilisez les 3 boutons du téléphone
- Regardez la différence avec et sans CLEAR_TOP et SINGLE_TOP

Important



Configurez le mode debug pour ne pas lancer l'Activity après le build.
Lancez l'Activity à la main, via le launcher.

Introduction aux interfaces graphiques





Création d'une interface graphique

- Dans des fichiers XML
- Chaque balise XML (layout, view) correspond à une classe Java
- Plusieurs Layouts en natif dans Android :
 - **AbsoluteLayout** (deprecated)
 - **FrameLayout**
 - **LinearLayout**
 - **RelativeLayout**
 - **TableLayout**



Notions de taille

- Chaque View doit posséder 2 attributs :

- layout_width
 - layout_height

- Valeurs possibles :

- **dp (dip)**
 - px, mm, in
 - **match_parent**
 - **wrap_content**

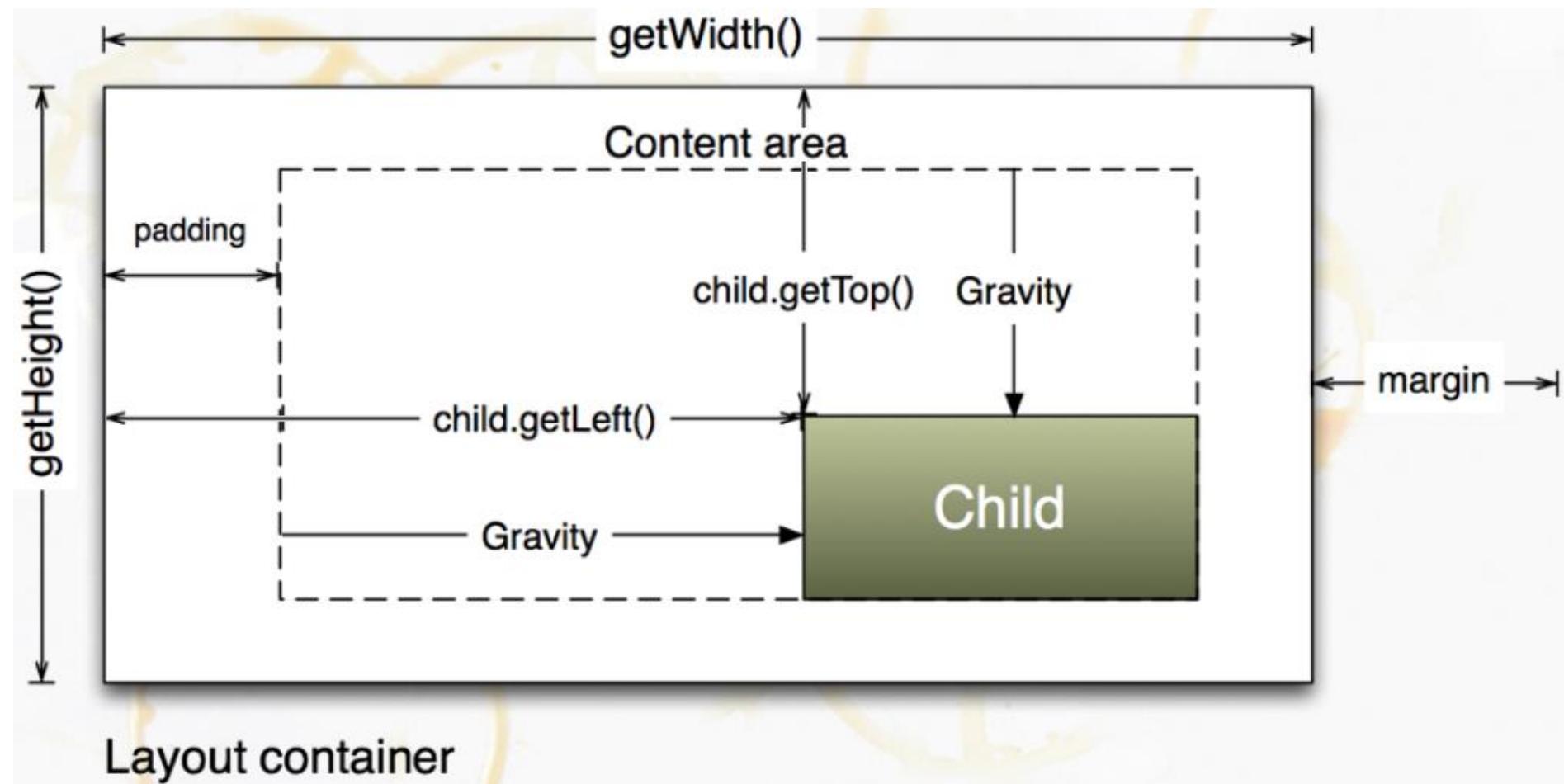
- Density-independant pixel

- 1 dp = 1 pixel sur un écran de densité 160 dpi (mdpi)
 - Conversion en pixels au runtime : $px = dp * (dpi / 160)$





Autopsie d'un Layout





Attributs XML

Attribut XML qui commence par **layout_**



N'affecte que l'élément qui déclare cet attribut

```
<LinearLayout  
    android:gravity="bottom"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

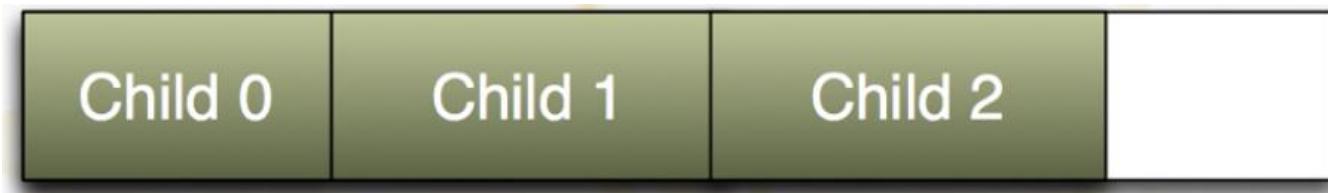
```
    <View  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_gravity="right" />
```

```
</LinearLayout>
```



LinearLayout

- Vertical ou horizontal





LinearLayout

- Remplissage de l'espace restant : layout_weight



No weight



Child2:
android:layout_weight="1.0"



Child1, Child2:
android:layout_weight="1.0"

```
<View android:layout_width="match_parent"  
      android:layout_height="0dip"  
      android:layout_weight="1" />
```



RelativeLayout

- Placer les éléments les uns par rapport aux autres



```
<RelativeLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">  
  
<View android:id="@+id/child0"  
      android:layout_width="wrap_content"  
      android:layout_height="wrap_content" />  
  
<View  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/child0"  
    android:layout_toRightOf="@id/child0"  
    android:layout_alignParentRight="true" />  
...  
</RelativeLayout>
```



RelativeLayout

- Les attributs de positionnement prennent sur les dimensions
- Donc : layout_width et layout_height peuvent être ignorés



```
<RelativeLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">  
  
<View android:id="@+id/child0"  
      android:layout_width="wrap_content"  
      android:layout_height="wrap_content" />  
  
<View  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/child0"  
    android:layout_toRightOf="@id/child0"  
    android:layout_alignParentRight="true" />  
...  
</RelativeLayout>
```



TableLayout

- Un layout, pas un widget
 - Pas de tri par colonne, etc
- Un ensemble de TableRow

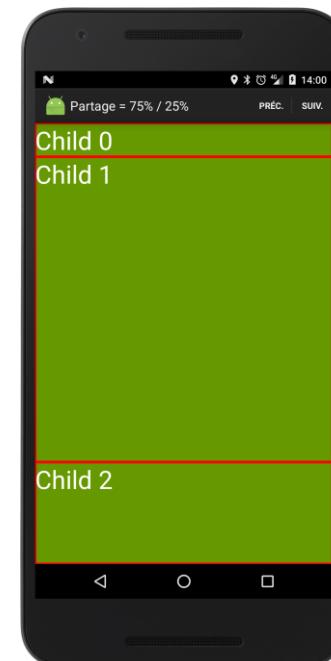
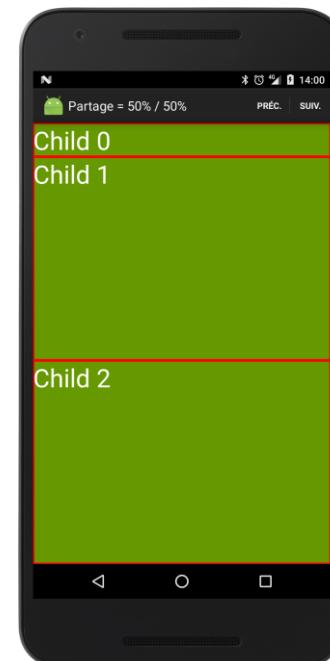
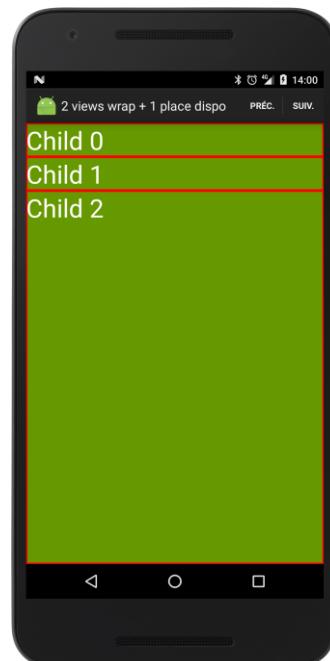
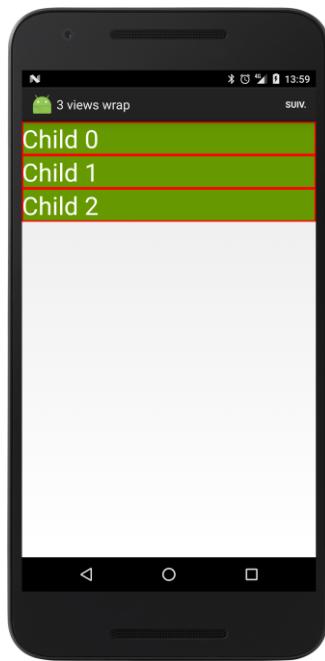
Child 0	Child 1	Child 2
Child 3	Child 4	Child 5
Child 6		Child 7

TD2 : création d'une première interface graphique

**TD disponible sur
<https://goo.gl/sFz8uJ>**



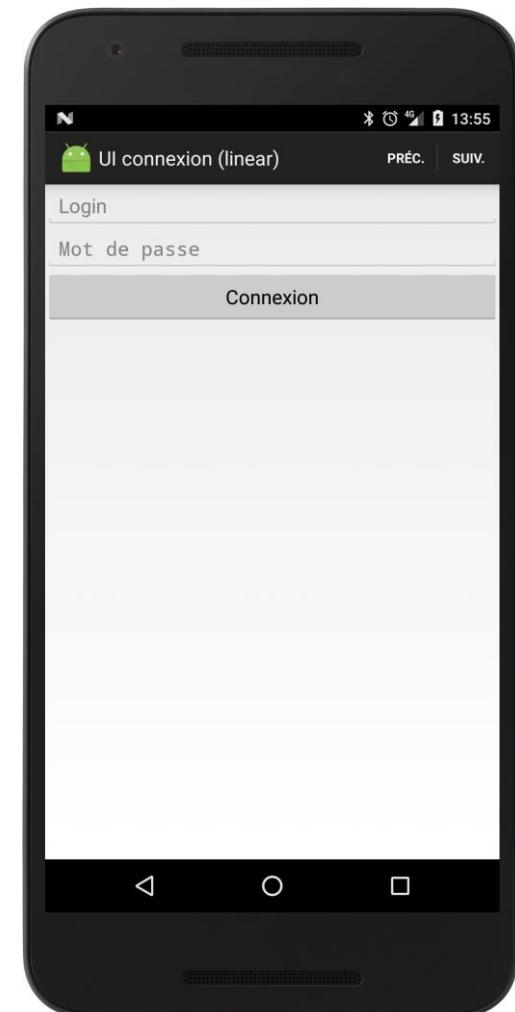
- Regardez les différentes vues
- Essayez de trouver comment réaliser chaque vue
- Solution dans le code





Exemple

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:padding="10dp">  
  
</LinearLayout>
```





Exemple

```
<LinearLayout ...>
```

```
  <EditText  
    android:id="@+id/login"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="@string/login"  
    android:inputType="numberSigned">
```

```
    <requestFocus />  
  </EditText>
```

```
</LinearLayout>
```





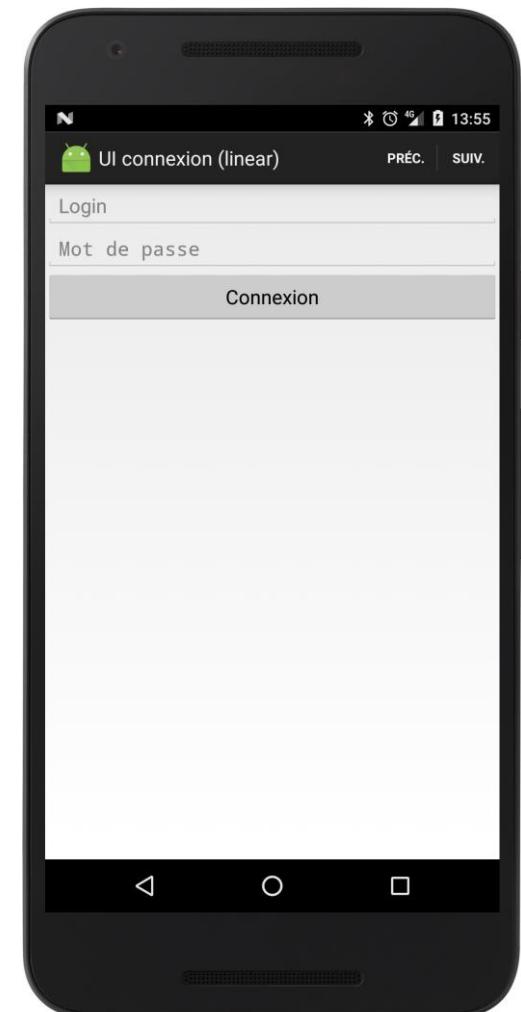
Exemple

```
<LinearLayout ...>
```

```
    <EditText ...></EditText>
```

```
    <EditText  
        android:id="@+id/password"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:hint="@string/password"  
        android:inputType="textPassword"  
        android:imeOptions="actionGo" />
```

```
</LinearLayout>
```





Exemple

```
<LinearLayout ...>
```

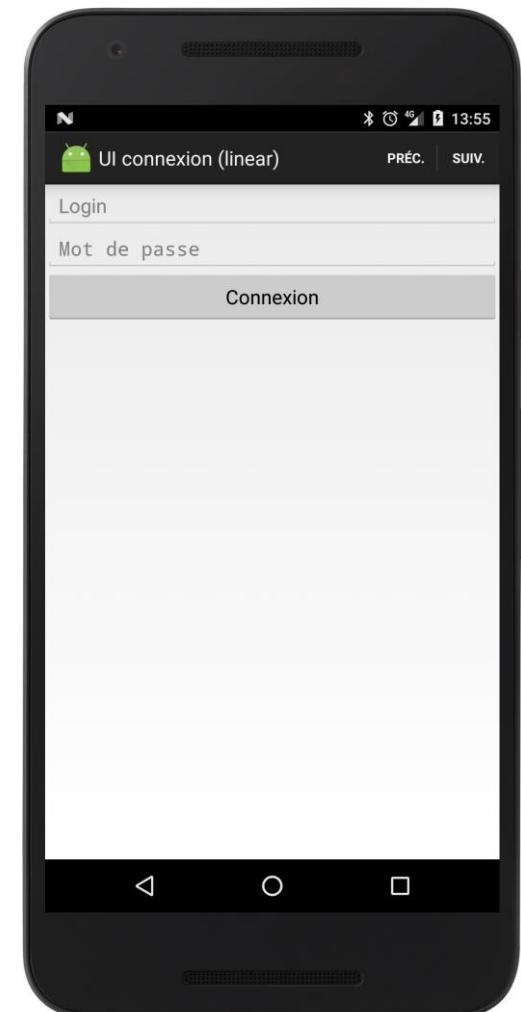
```
    <EditText ...></EditText>
```

```
    <EditText .../>
```

```
    <Button
```

```
        android:id="@+id/loginBtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/loginBtn" />
```

```
</LinearLayout>
```



TP1 : Affichage des lieux dans une liste + carte Google Map

TP2

Création d'une calculatrice

Les bases de données





SGBD : SQLite

- Manipulation avec le langage SQL
- Base directement intégrée au programme (pas de client – serveur)
- Pourquoi SQLite ?
 - Légereté : environ 300Ko
 - Très faible conso mémoire au runtime
- Licence : domaine public



SQLite : Types de données

- TEXT
 - CHAR, VARCHAR, TEXT
- INTEGER
 - TINYINT, SMALLINT, INT, MEDIUMINT, BIGINT
- REAL
 - DOUBLE, FLOAT, REAL
- NUMERIC
 - NUMERIC, DECIMAL, BOOLEAN, DATE, DATETIME



SQLiteOpenHelper

- Helper qui aide à gérer plusieurs cas :
 - Tout 1^{er} lancement après l'installation : CREATE DATABASE
 - Mise à jour de la structure de la base
- 2 méthodes à redéfinir :

abstract void	<code>onCreate(SQLiteDatabase db)</code> Called when the database is created for the first time.
abstract void	<code>onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)</code> Called when the database needs to be upgraded.



SQLiteOpenHelper

- Exemple

```
public class DatabaseHelper extends SQLiteOpenHelper
{
    private static final String DATABASE_NAME = "wellbenet.db";
    private static final int SCHEMA_VERSION = 3;
    private static DatabaseHelper singleton;

    public static synchronized DatabaseHelper getInstance(Context context)
    {
        if (singleton == null)
            singleton = new DatabaseHelper(context.getApplicationContext());
        return singleton;
    }

    private DatabaseHelper(Context context)
    {
        super(context, DATABASE_NAME, null, SCHEMA_VERSION);
    }

    [...]
}
```



SQLiteOpenHelper

- onCreate

```
@Override  
public void onCreate(SQLiteDatabase db)  
{  
    try  
    {  
        db.beginTransaction();  
  
        FirstTable.onCreate(db);  
        SecondTable.onCreate(db);  
  
        db.setTransactionSuccessful();  
    }  
    finally  
    {  
        db.endTransaction();  
    }  
}
```



SQLiteOpenHelper

- `onUpgrade`
- Exemple :
 - Version nouvellement installée : 3
 - Ancienne version : 1
- L'utilisateur ne passe pas forcément par la version 2 !



Requête sur la BDD

- Adapter très pratique : SimpleCursorAdapter
 - Layout d'une ligne
 - Colonnes requêtées à afficher
 - Identifiants des vues dans lesquelles afficher les données

```
SimpleCursorAdapter(Context context, int layout, Cursor c, String[] from, int[] to, int flags)
```

Standard constructor.

- Exemple :

```
SELECT _id, category, day_duration, week_frequency FROM objectives
```

```
String[] columns = new String[] { "category", "day_duration", "week_frequency" };  
int[] to = new int[] { R.id.category, R.id.duration, R.id.frequency };
```

```
adapter = new SimpleCursorAdapter(activity, R.layout.objective_row, null, columns, to, 0);  
myListView.setAdapter(adapter);
```



Comment faire une requête ?

- Best practices :
 - Requêtes asynchrones
 - Utiliser un ContentProvider
- Implémenter `LoaderManager.LoaderCallbacks<Cursor>`

Public Methods

abstract `Loader<D>`

`onCreateLoader(int id, Bundle args)`

Instantiate and return a new Loader for the given ID.

abstract void

`onLoadFinished(Loader<D> loader, D data)`

Called when a previously created loader has finished its load.

abstract void

`onLoaderReset(Loader<D> loader)`

Called when a previously created loader is being reset, and thus making its data unavailable.



Les ContentProviders

- Rappel : Encapsulation de données (fichiers, BDD, Internet...)
- Accès par une URI (`content://`)
 - URI de base : collection d'objets
`content://contacts/people`
 - URI d'une instance particulière
`content://contacts/people/5`
- Opérations CRUD
 - Create `insert() : Uri`
 - Read `query() : Cursor`
 - Update `update() : int`
 - Delete `delete() : int`



ContentProvider : pour aller plus loin

- Autre méthodes disponibles :

- bulkInsert () insertions multiples

Attention : insertion non atomique.

Si besoin, surcharger la méthode pour y ajouter les transactions.

- applyBatch () opérations multiples

ContentProviderResult[]	<code>applyBatch(String authority, ArrayList<ContentProviderOperation> operations)</code> Applies each of the <code>ContentProviderOperation</code> objects and returns an array of their results.
final int	<code>bulkInsert(Uri url, ContentValues[] values)</code> Inserts multiple rows into a table at the given URL.



Utilisation d'un ContentProvider

- A partir du ContentResolver
 - Trouve le bon ContentProvider à partir de son URI

```
Cursor results = getContentResolver().query(MyContentProvider.MY_URI,  
                                         projection,  
                                         selection,  
                                         selectionArgs,  
                                         sortOrder);
```



Utilisation des Loaders

- `onCreateLoader()` : instantiation d'un Loader
 - définition de la requête

```
public Loader<Cursor> onCreateLoader(int id, Bundle bundle)
{
    String[] projection = new String[] {
        ObjectiveTable.COLUMN_ID, ObjectiveTable.COLUMN_CATEGORY,
        ObjectiveTable.COLUMN_DAY_DURATION, ObjectiveTable.COLUMN_WEEK_FREQUENCY
    };
    String selection = ObjectiveTable.COLUMN_ID_PROJECT + " = ?";
    String[] selectionArgs = new String[] { projectId };
    String sortOrder = ObjectiveTable.COLUMN_ID;

    return new CursorLoader(activity,
        ObjectiveProvider.OBJECTIVES_URI,
        projection,
        selection,
        selectionArgs,
        sortOrder);
}
```



Utilisation des Loaders

- onLoadFinished() : requête exécutée, résultats disponibles

```
public void onLoadFinished(Loader<Cursor> loader, Cursor cursor)
{
    adapter.swapCursor(cursor);

    TextView noDataView = (TextView)activity.findViewById(R.id.noData);
    noDataView.setVisibility(cursor.getCount() > 0 ? View.GONE : View.VISIBLE);
}
```



Utilisation des Loaders

- `onLoaderReset () :`
 - Appelée lors d'un reset du loader
 - Ne plus utiliser le curseur, car les données sont indisponibles

```
public void onLoaderReset(Loader<Cursor> loader)
{
    adapter.swapCursor(null);
}
```



Loaders : FAQ

- Comment démarrer le Loader ?

```
abstract <D> Loader<D>    initLoader(int id, Bundle args, LoaderCallbacks<D> callback)
```

Ensures a loader is initialized and active.

```
getLoaderManager().initLoader(LOADER_ID, null, this);
```

- Où fermer le curseur ??
 - Le Loader s'en charge ☺
- Que se passe-t-il si les données changent ?
 - CursorLoader créé à partir d'un ContentProvider :
onLoadFinished() rappelé lorsque les données changent

Attention : uniquement si l'URI utilisée pour un insert / update / delete est la même que celle utilisée par le CursorLoader