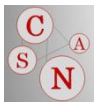




Διάλεξη «Εισαγωγή στην πλατφόρμα Android»

Εισηγητής: Παντελής Μπαλαούρας



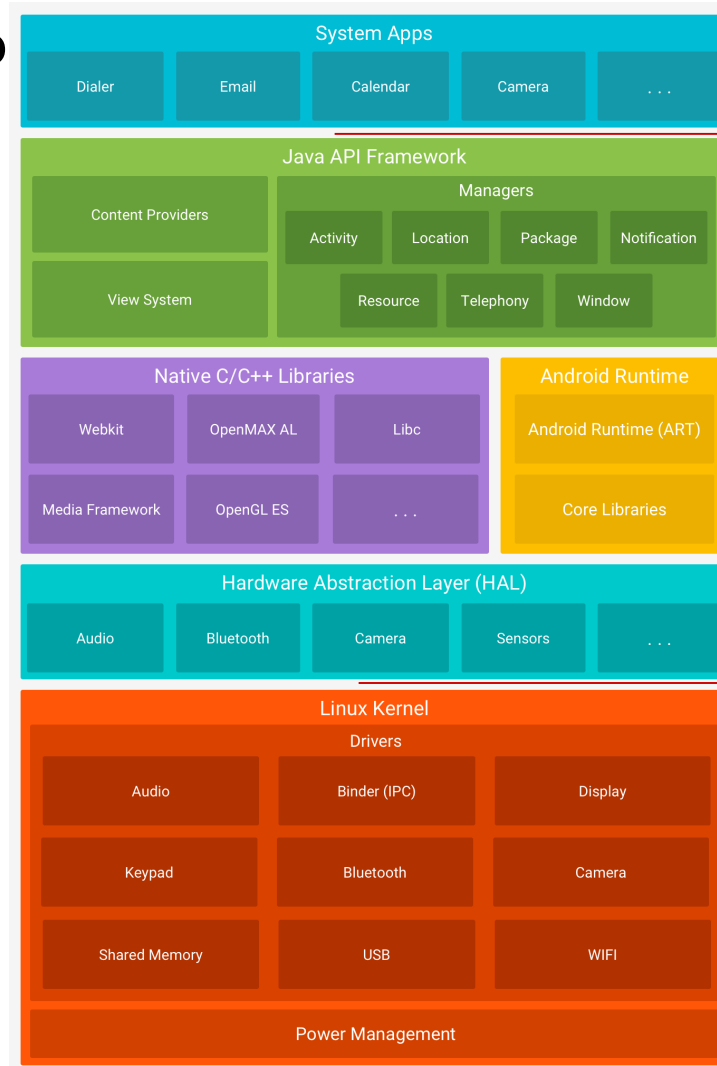
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΙΚΑ ΣΥΣΤΑΤΙΚΑ ANDROID

Τι είναι το Android

Ένα **Λειτουργικό Σύστημα**,

για **στοίβα λογισμικού** για κινητές συσκευές με

- χαμηλή επεξεργαστική ισχύ
- χαμηλή μνήμη
- οθόνη αφής ως μέσο διεπαφής
- ένας χρήστης



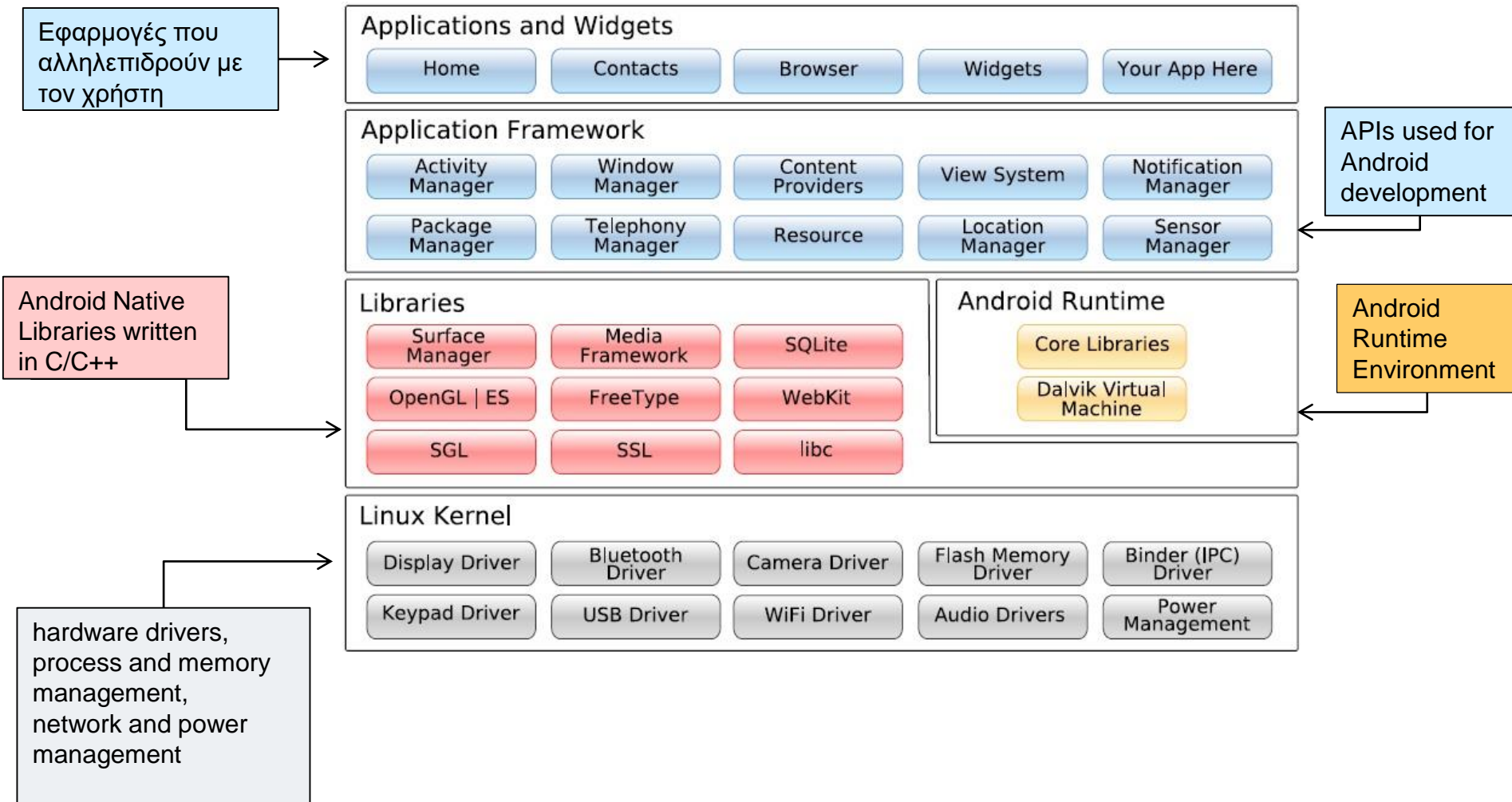
περιλαμβάνει

Βασικές εφαρμογές

Μεσισμικό (middleware)

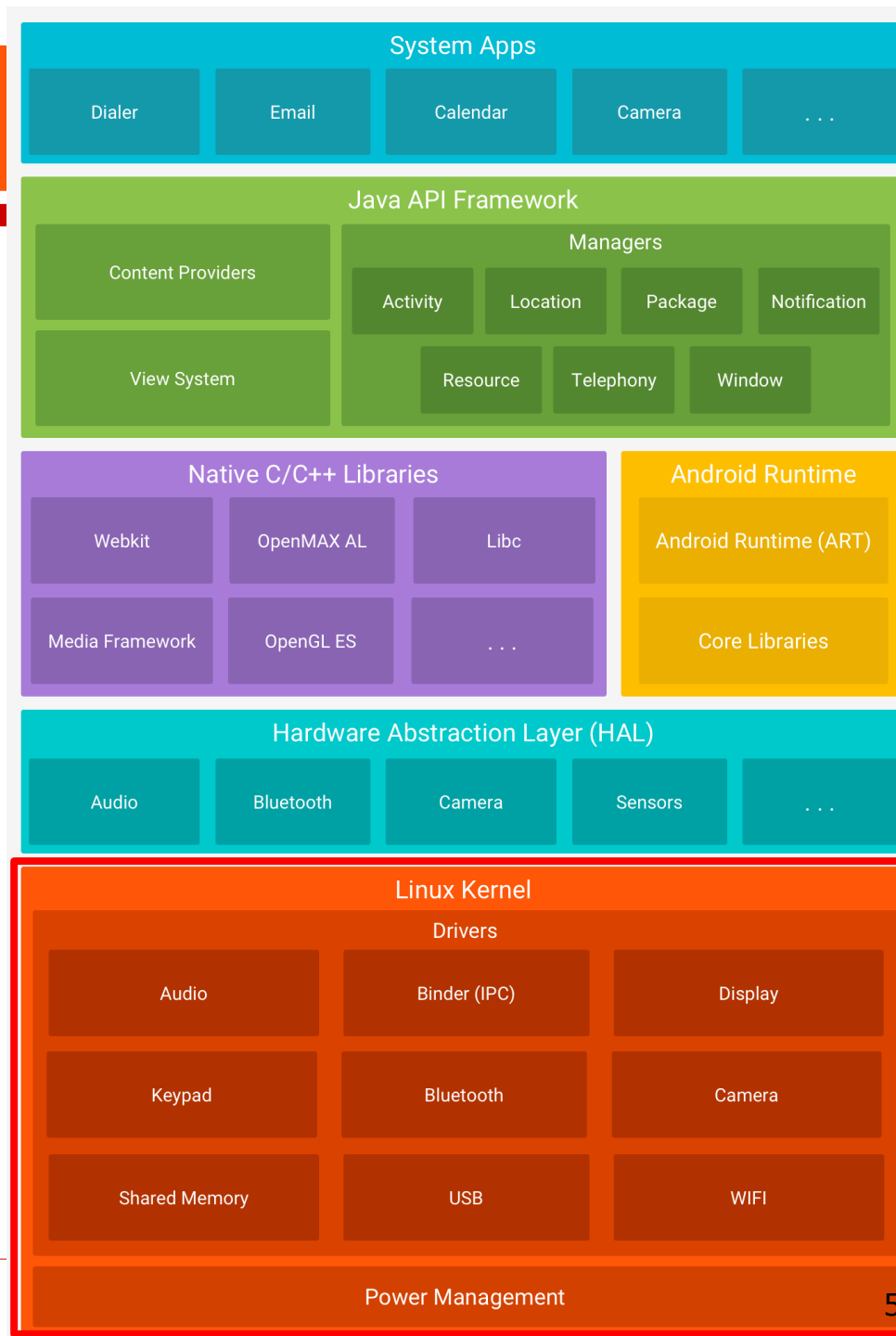
Λειτουργικό σύστημα

Αρχιτεκτονική Android



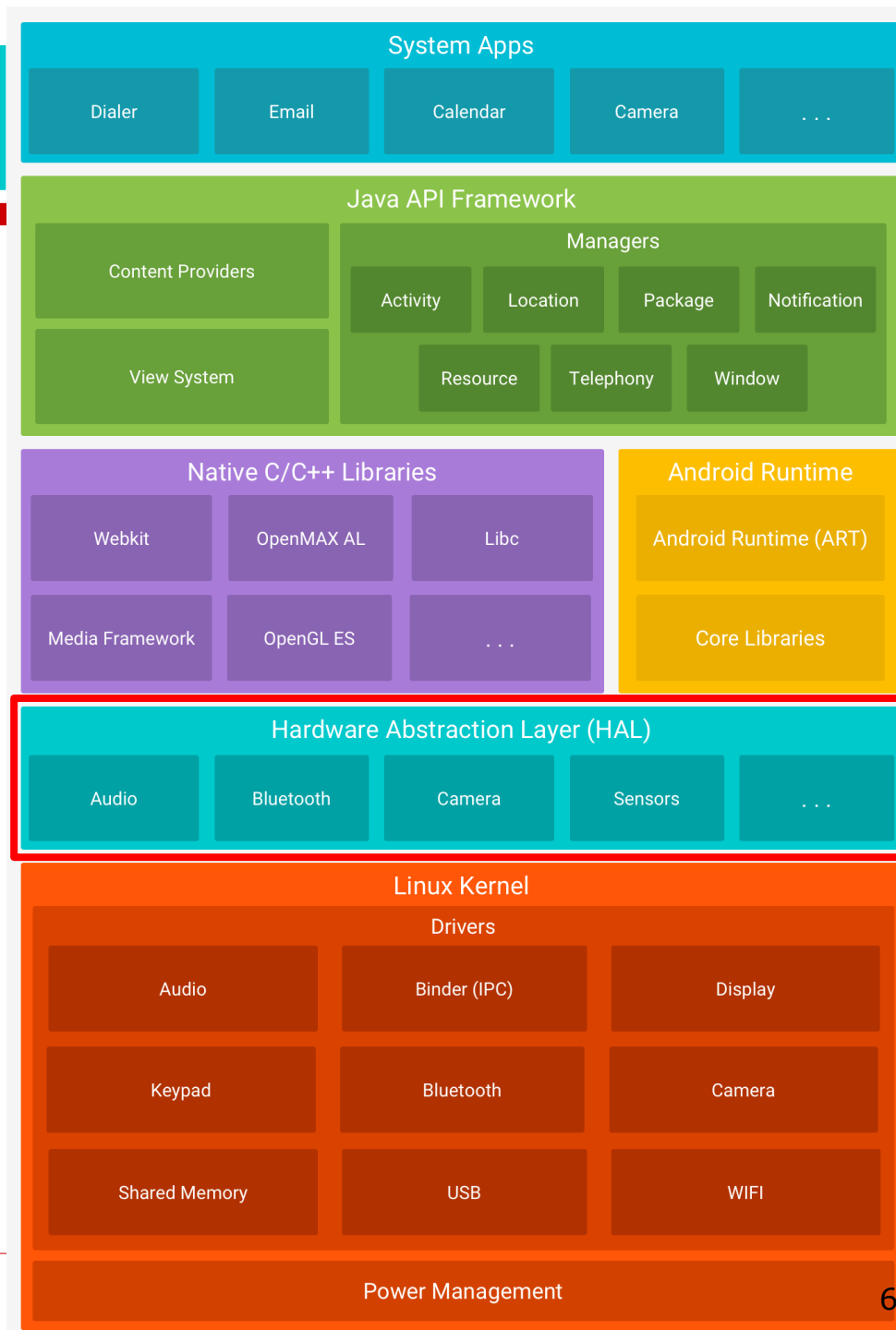
Linux kernel (LK)

- Είναι η βάση του Android π.χ. το Android Runtime (ART) βασίζεται στο Linux kernel για:
 - Διαχείριση μνήμης (low-level memory management)
 - Διαχείριση διεργασιών και νημάτων
 - Διαχείριση ενέργειας
- Επιτρέπει στο Android να εκμεταλλευτεί βασικά χαρακτηριστικά ασφάλειας του Linux Kernel
- Επιτρέπει στους κατασκευαστές να αναπτύξουν drivers σε ένα γνωστό και καλά ορισμένο kernel
- Διαδεδομένο και δοκιμασμένο ΛΣ ανοικτού κώδικα με κατάλληλη άδεια χρήσης για το business model των κατασκευαστών κινητών συσκευών



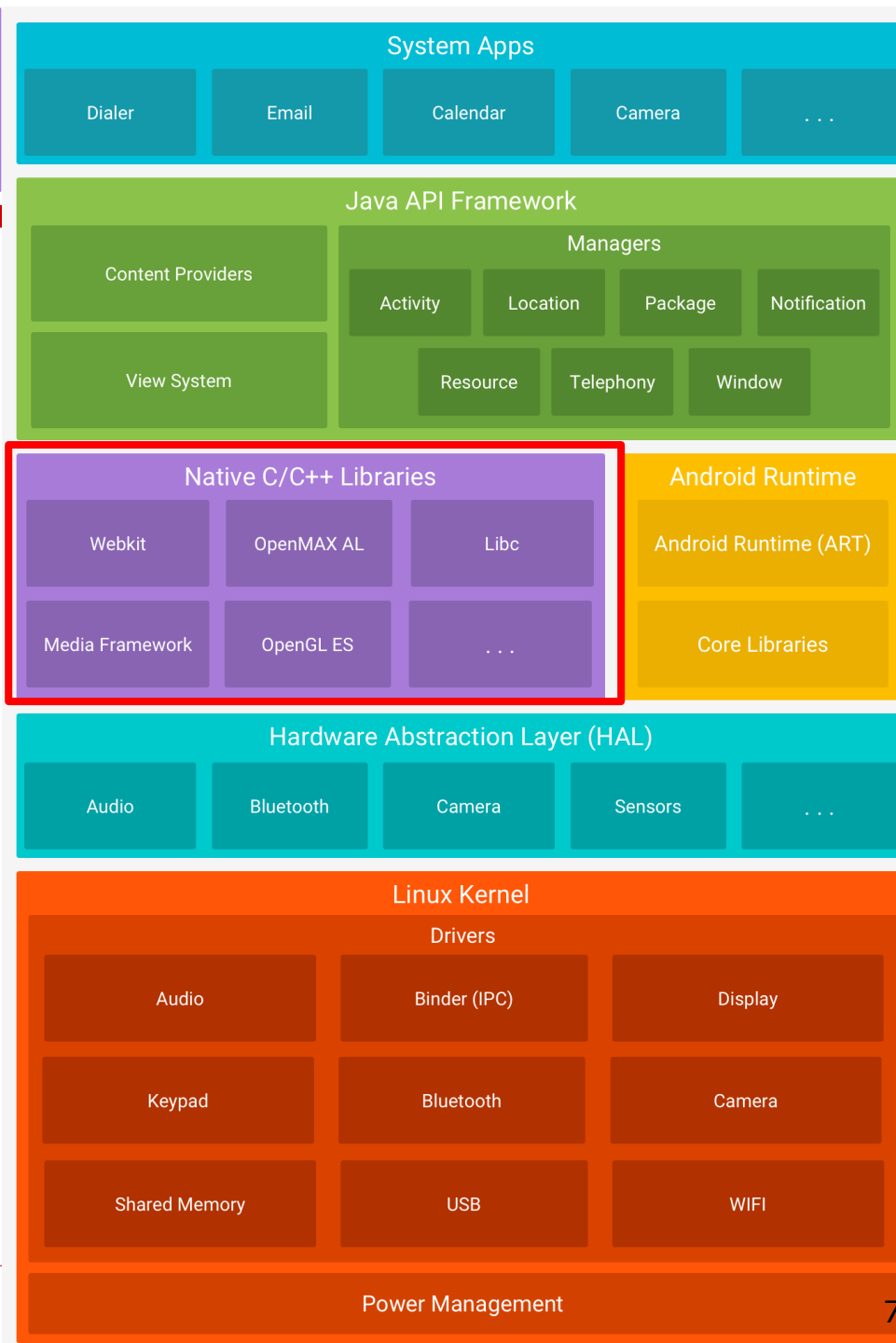
Hardware Abstraction Layer (HAL)

- Παρέχει διεπαφές (standard interfaces) για χρήση των δυνατοτήτων των συστατικών **hardware** από το Java API framework
- Πολλαπλές βιβλιοθήκες, κάθε μία υλοποιεί μία συγκεκριμένη διεπαφή για ένα τύπο συστατικού hardware
- Όταν ένα framework API καλεί για χρήση μία συσκευή, το Android φορτώνει τη βιβλιοθήκη για αυτή τη συσκευή
- Απομονώνει το LK από το API Framework



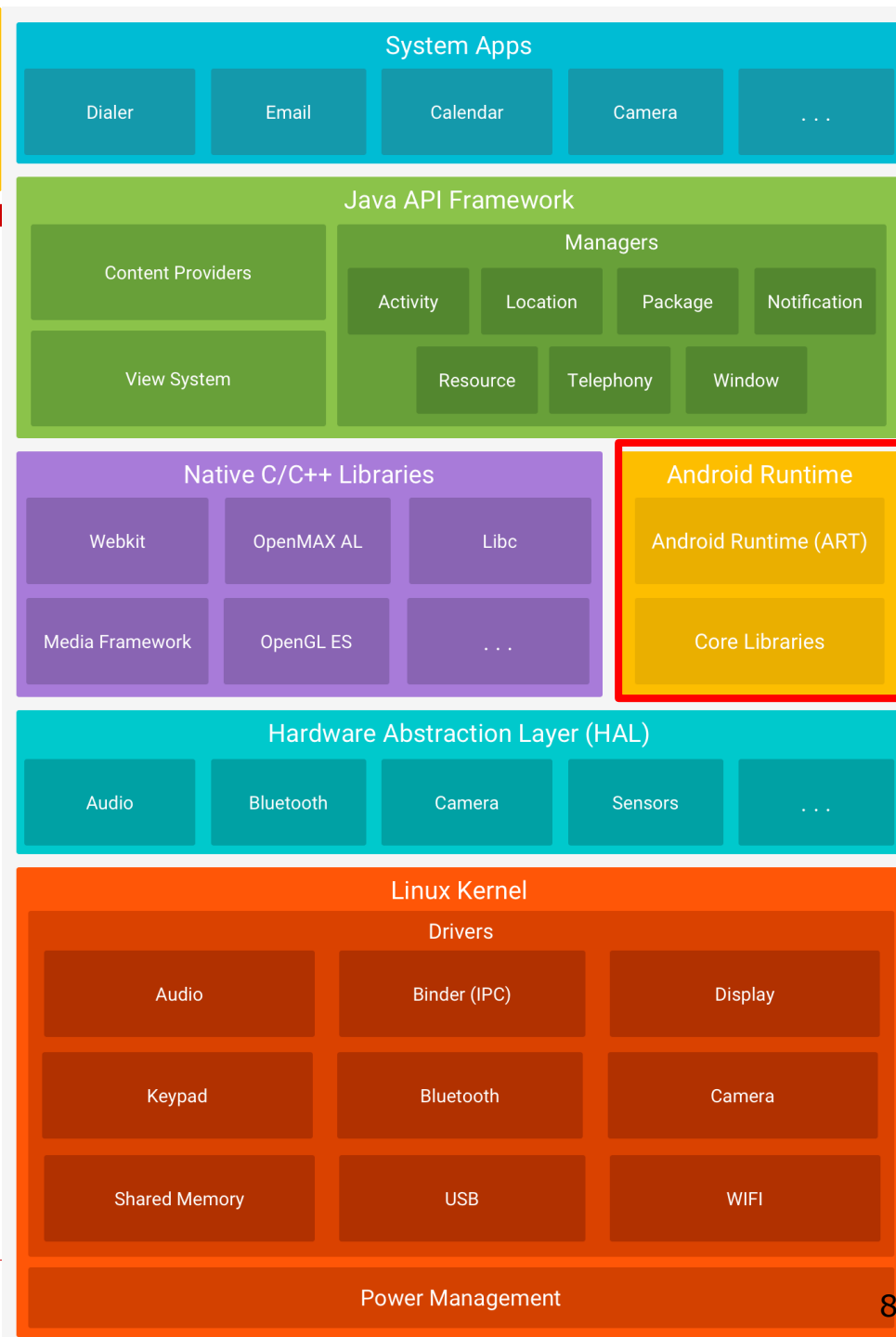
Native C/C++ Libraries

- Πολλά συστατικά του Android system, π.χ., ART και HAL, αναπτύσσονται με βάση κώδικα και απαιτούν βιβλιοθήκες γραμμένες σε C και C++.
- Το Android παρέχει Java framework APIs για να διαθέτει τη λειτουργικότητα των libraries στα apps
 - π.χ. χρήση **OpenGL ES** μέσω του Android framework's Java OpenGL API για το **σχεδιασμό και χειρισμό γραφικών 2D και 3D σε μία εφαρμογή.**



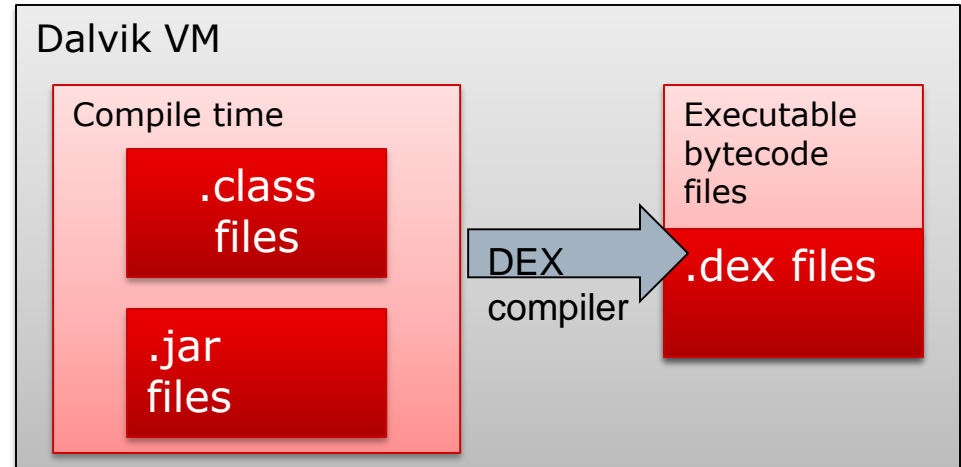
Android Runtime Environment

- Τρέχει το εκτελέσιμο κώδικα των εφαρμογών
- Dalvik ή ART για version 5.0+ (API level 21+)
- Κάθε app τρέχει ως διακριτή διεργασία και σε δικό του στιγμιότυπο (instance) του ART.
- Το ART (και το Dalvik) τρέχει σε πολλαπλά virtual machines σε συσκευές χαμηλής μνήμης εκτελώντας DEX files (των apps)



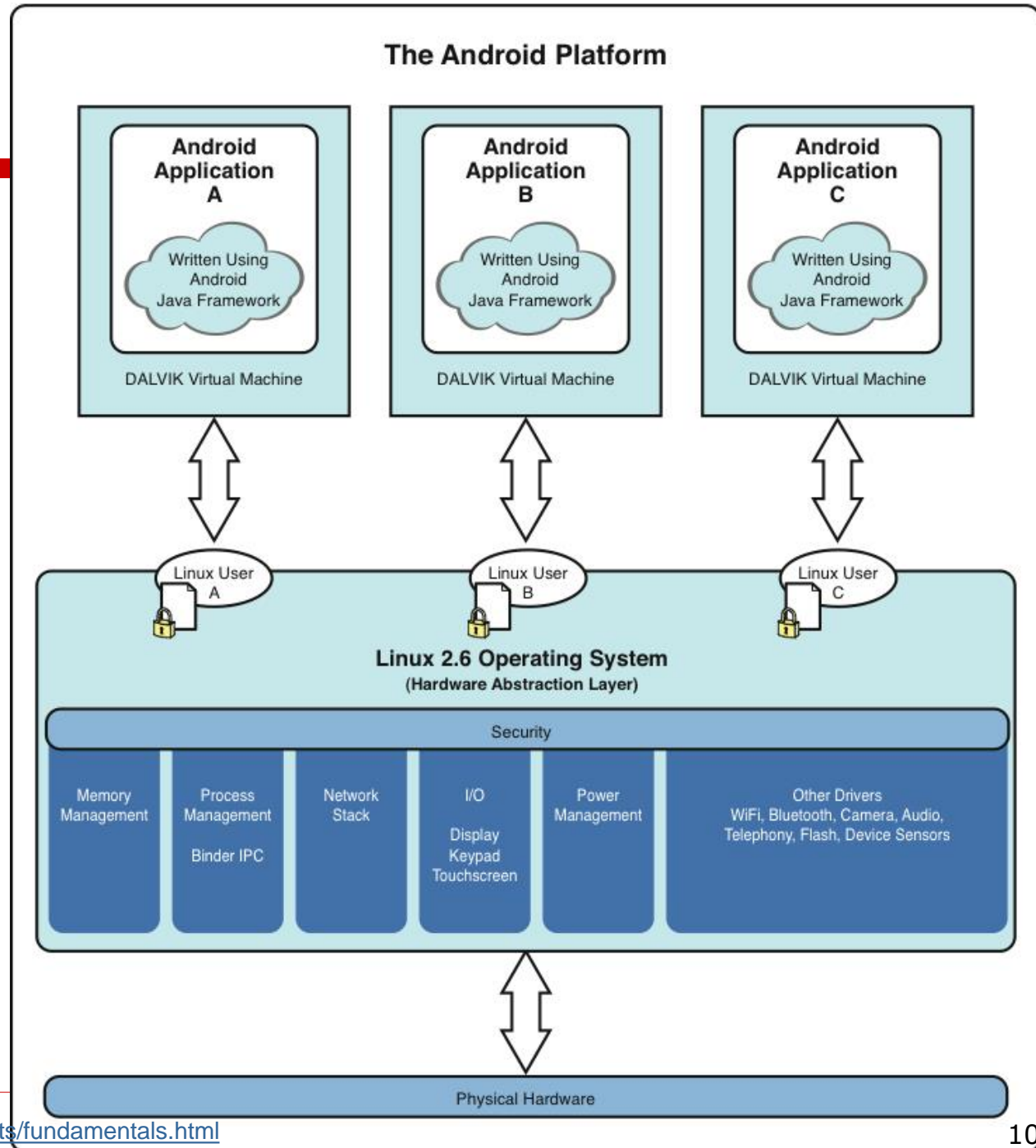
Dalvik Virtual Machine (VM)

- Βελτιστοποιημένο Java VM με χαμηλές απαιτήσεις μνήμης
- Μεταγλώττιση κώδικα σε machine-independent εντολές (bytecodes) οι οποίες εκτελούνται από το Dalvik VM στη κινητή συσκευή
- Just-in-time (JIT) compilation
 - Ο DEX JIT compiler (dx tool) μεταγλωττίζει τα Java bytecode σε native machine language κατά την εκτέλεση του κώδικα (run time) και όχι πρωτύτερα
- Βασίζεται στο Linux Kernel για
 - Threading
 - Low-level διαχείριση μνήμης
- Δυνατότητα πολλαπλών VM στιγμιοτύπων που επωφελούνται από το Linux OS για απομόνωση ασφάλειας και διεργασιών



Αρχιτεκτονική Android

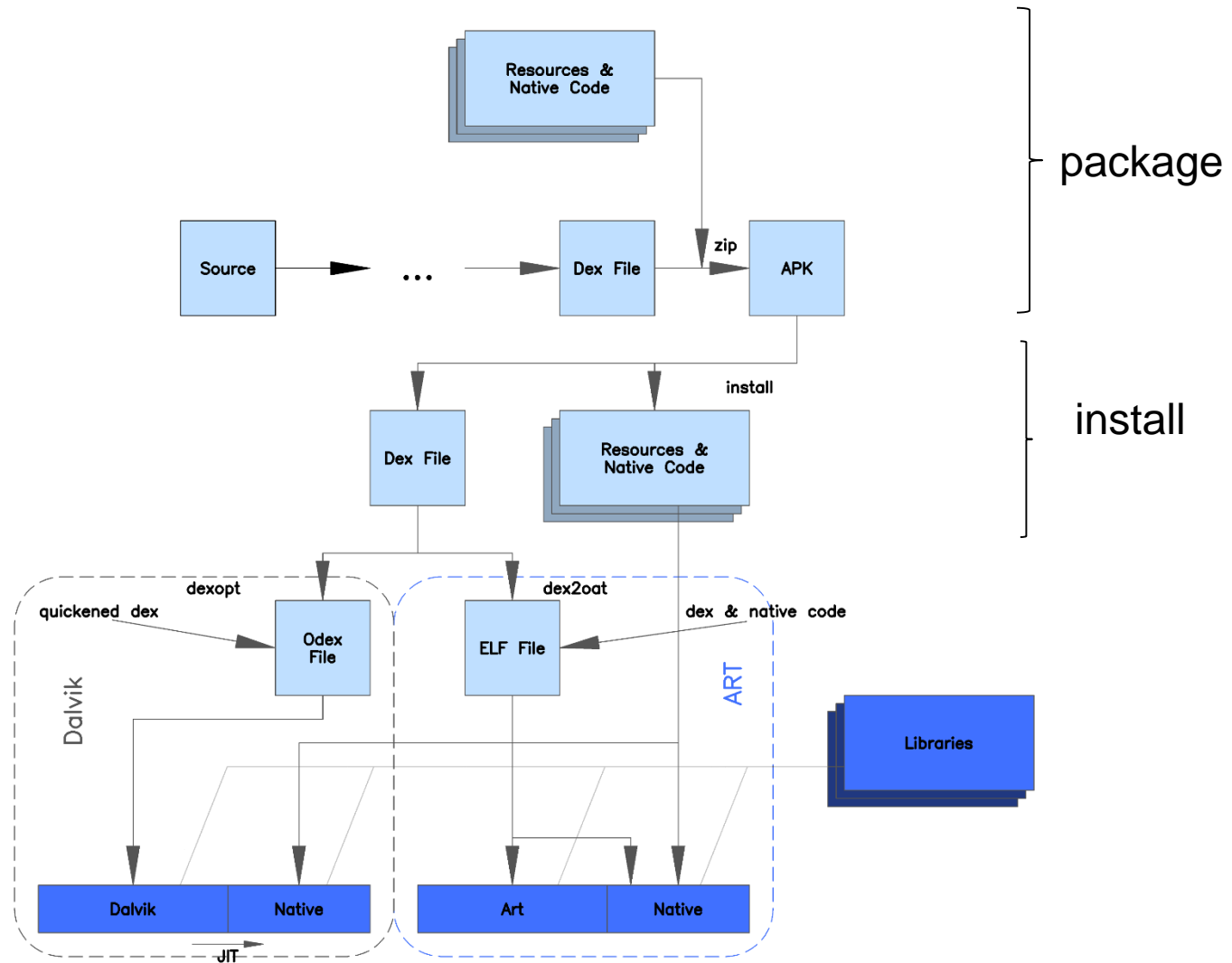
- Κάθε εφαρμογή τρέχει στη δική της διαδικασία Linux
- Κάθε διαδικασία έχει τη δική της Dalvik VM
- Σε κάθε εφαρμογή δίνεται ένα μοναδικό Linux user ID
- Τα Permissions καθορίζονται ώστε μια εφαρμογή να έχει πρόσβαση σε συγκεκριμένα αρχεία



Android RunTime (ART) αντί Dalvik

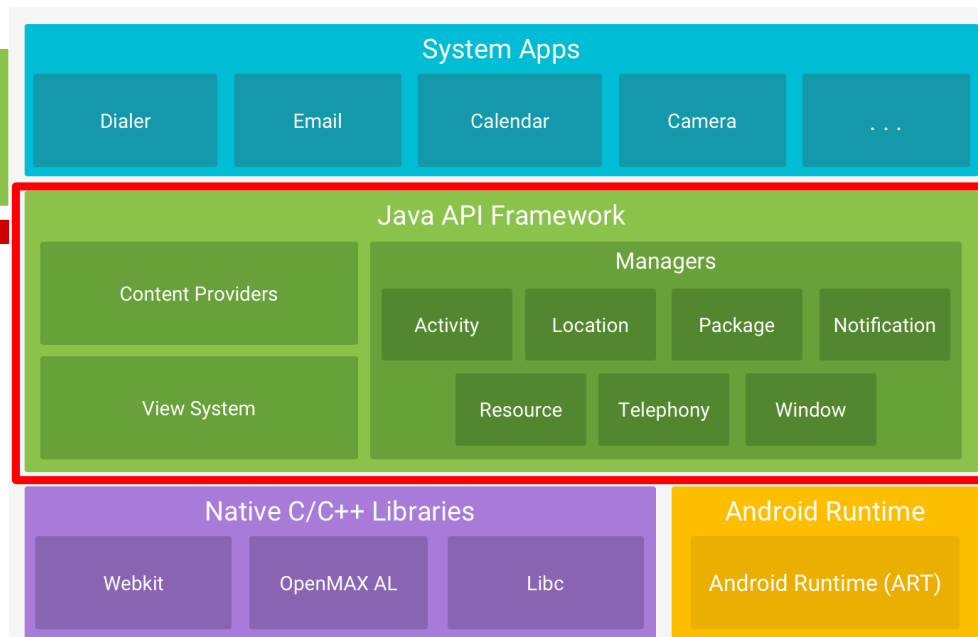
- Default από Android Lollipop (5.0)
- Ahead Of Time (AOT) compilation αντί JIT compilation
 - AOT μετατρέπει το bytecode ενός VM σε machine code πριν την εκτέλεση (κατά την εγκατάσταση της εφαρμογής με το dex2oat tool)
 - Καλύτερη απόδοση
 - Μικρότερη κατανάλωση ενέργειας
 - Καλύτερο garbage collection
- Καλύτερο περιβάλλον ανάπτυξης και debugging
- ART και το Dalvik είναι συμβατά, έτσι apps που αναπτύχθηκαν για το Dalvik θα πρέπει να τρέχουν και με ART

ART vs Dalvik

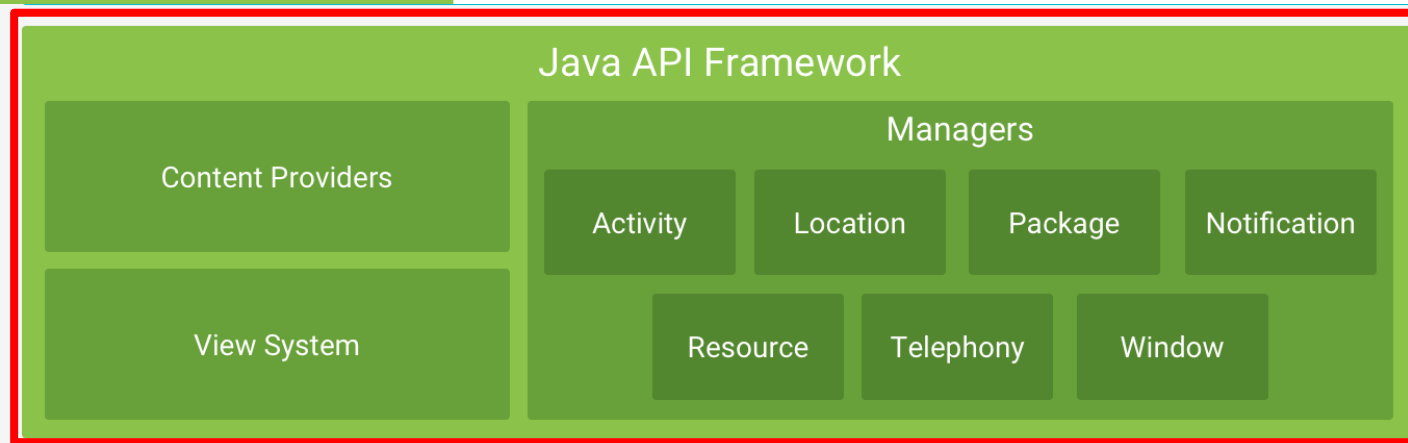


Java API Framework

- Όλο το set των δυνατοτήτων & χαρακτηριστικών του Android OS είναι διαθέσιμο στον προγραμματιστή μέσω APIs γραμμένα σε Java
- Τα APIs είναι τα βασικά building blocks που χρειάζεται ένας προγραμματιστής για να δημιουργήσει Android εφαρμογές (apps) επαναχρησιμοποιώντας
 - βασικά και δομικά συστατικά
 - υπηρεσίες
- Οι προγραμματιστές έχουν πλήρη πρόσβαση στο ίδιο Developers framework APIs που χρησιμοποιούν οι Android system apps

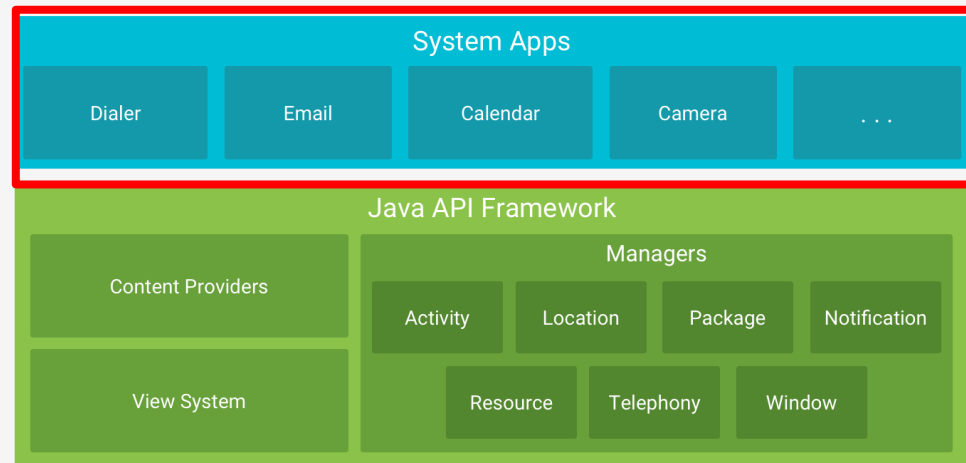


Java API



- Κυριότερα δομικά συστατικά & υπηρεσίες:
 - **View System** για την ανάπτυξη της διεπαφής χρήσης της εφαρμογής (app's UI) π.χ., lists, grids, text boxes, buttons embeddable web browser
 - **Content Providers:** επιτρέπει στις εφαρμογές να έχουν πρόσβαση σε δεδομένα από άλλες εφαρμογές (π.χ. εφαρμογή επαφών) ή να μοιράζουν τα δικά τους δεδομένα
 - Ο **Resource Manager** παρέχει πρόσβαση σε πόρους, **αρχεία πλην των αρχείων κώδικα**, π.χ. τοπικές συμβολοσειρές (localized strings), γραφικά (graphics), και αρχεία που ορίζουν τη διάταξη (layout files)
 - Ο **Notification Manager** επιτρέπει σε όλες τις εφαρμογές να εμφανίζουν μηνύματα στη γραμμή κατάστασης (custom alerts in the status bar)
 - Ο **Activity Manager** που διαχειρίζεται το κύκλο ζωής των εφαρμογών και παρέχει την κοινή στοίβα πλοήγησης από εφαρμογή σε εφαρμογή με το back (navigation back stack)

Εφαρμογές συστήματος (System Apps)



- Το Android συνοδεύεται από ένα σύνολο από βασικές εφαρμογές: email, SMS messaging, calendars, internet browsing, contacts, κ.α.
- Αυτές δεν έχουν κάποια διάκριση σε σχέση με αυτές που επιλέγει να εγκαταστήσει ο χρήστης.
 - Έτσι, εφαρμογές τρίτων (third-party app) μπορούν να επιλεγούν ως οι προκαθορισμένες (default) από το χρήστη για web browser, SMS messenger, πληκτρολόγιο.
 - Εξαιρούνται οι ρυθμίσεις.
- Επιπλέον, παρέχουν βασικές δυνατότητες που μπορούν να χρησιμοποιήσει ένας προγραμματιστής σε μια νέα εφαρμογή. Π.χ. εάν η εφαρμογή σας θέλετε να στείλει ένα SMS, δεν απαιτείται να αναπτύξετε εξ αρχής τη λειτουργικότητα αλλά να καλέσετε την SMS app που είναι ήδη εγκατεστημένη.

Βασικά Δομικά Συστατικά Android

□ **Activity**

- Το στρώμα παρουσίασης της εφαρμογής
- Μια εφαρμογή μπορεί να έχει περισσότερα του ενός activities για διαχείριση διαφορετικών φάσεων του προγράμματος
- Κάθε activity είναι υπεύθυνο να αποθηκεύει την δική του κατάσταση

□ **Intent**

- Μηχανισμός που καθορίζει ποια συγκεκριμένη ενέργεια (action) πρέπει να εκτελεστεί
- Σε android σχεδόν ΟΛΕΣ οι αλληλοεπιδράσεις γίνονται μέσω intents

□ **Service**

- Διεργασία που τρέχει στο παρασκήνιο χωρίς τη παρέμβαση του χρήστη.
- Δεν παρέχει UI στο χρήστη
- Παρόμοιο με Unix daemon
- Υπάρχουν πολλά built-in services σε Android

□ **Content Providers**

- Επιτρέπουν σε μια εφαρμογή να έχει πρόσβαση σε δεδομένα άλλων εφαρμογών (π.χ. Contacts)
- Επιτρέπουν σε μια εφαρμογή να διαμοιράζει τα δεδομένα που παράγει σε άλλες εφαρμογές

ΔΟΜΙΚΑ ΣΥΣΤΑΤΙΚΑ ΕΦΑΡΜΟΓΩΝ (APPS)

Βασικά δομικά συστατικά εφαρμογών (App components)

- Τι είναι τα App components
 - είναι **βασικά δομικά συστατικά** μιας εφαρμογής
 - σημείο εισόδου προς την εφαρμογή σας από
 - το σύστημα Android
 - ένα χρήστη
- Τέσσερις (4) διαφορετικοί τύποι συστατικών:
 - Activities
 - Services
 - Broadcast receivers
 - Content providers

Βασικά δομικά συστατικά εφαρμογών (App components)

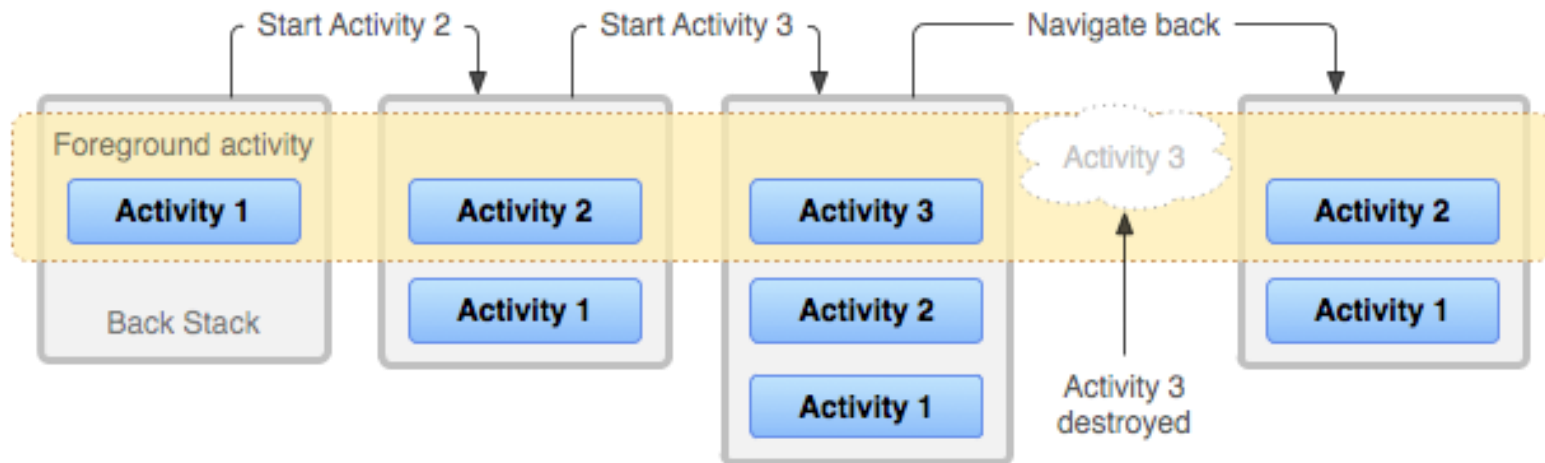
- Κάθε τύπος συστατικού
 - εξυπηρετεί ένα διακριτό σκοπό
 - έχει ένα διακριτό κύκλο ζωής που ορίζει το πώς κάθε συστατικό
 - δημιουργείται (created)
 - καταστρέφεται (destroyed)
- Μία εφαρμογή αποτελείται από ένα σύνολο συστατικών
 - **Μία** ή περισσότερες activities
 - Καμία, μία ή περισσότερες
 - Services
 - Broadcast receivers
 - Content providers

Activity ένα συστατικό εφαρμογής (1)

- Αφορά το στρώμα παρουσίασης (**presentation layer**) της εφαρμογής.
 - Αντιστοιχεί σε **μία οθόνη (screen)** της εφαρμογής
- Είναι το σημείο εισόδου για διάδραση με το χρήστη
- Μια εφαρμογή μπορεί να έχει περισσότερα του ενός activities για διαχείριση διαφορετικών οθονών της εφαρμογής
 - π.χ. μία εφαρμογή email μπορεί να έχει μία activity που δείχνει τη λίστα με νέα emails, μία άλλη για τη σύνθεση email, μία τρίτη για την ανάγνωση των emails κ.α.
- **Παρόλου που παρέχουν μία ενιαία εμπειρία χρήσης, κάθε activity είναι ανεξάρτητη από την άλλη**
- Μια άλλη εφαρμογή μπορεί να καλέσει και ξεκινήσει μία από αυτές τις activities (σημείο εισόδου)
 - Επαναχρησιμοποιήσιμο δομικό συστατικό

Activity ένα συστατικό εφαρμογής (2)

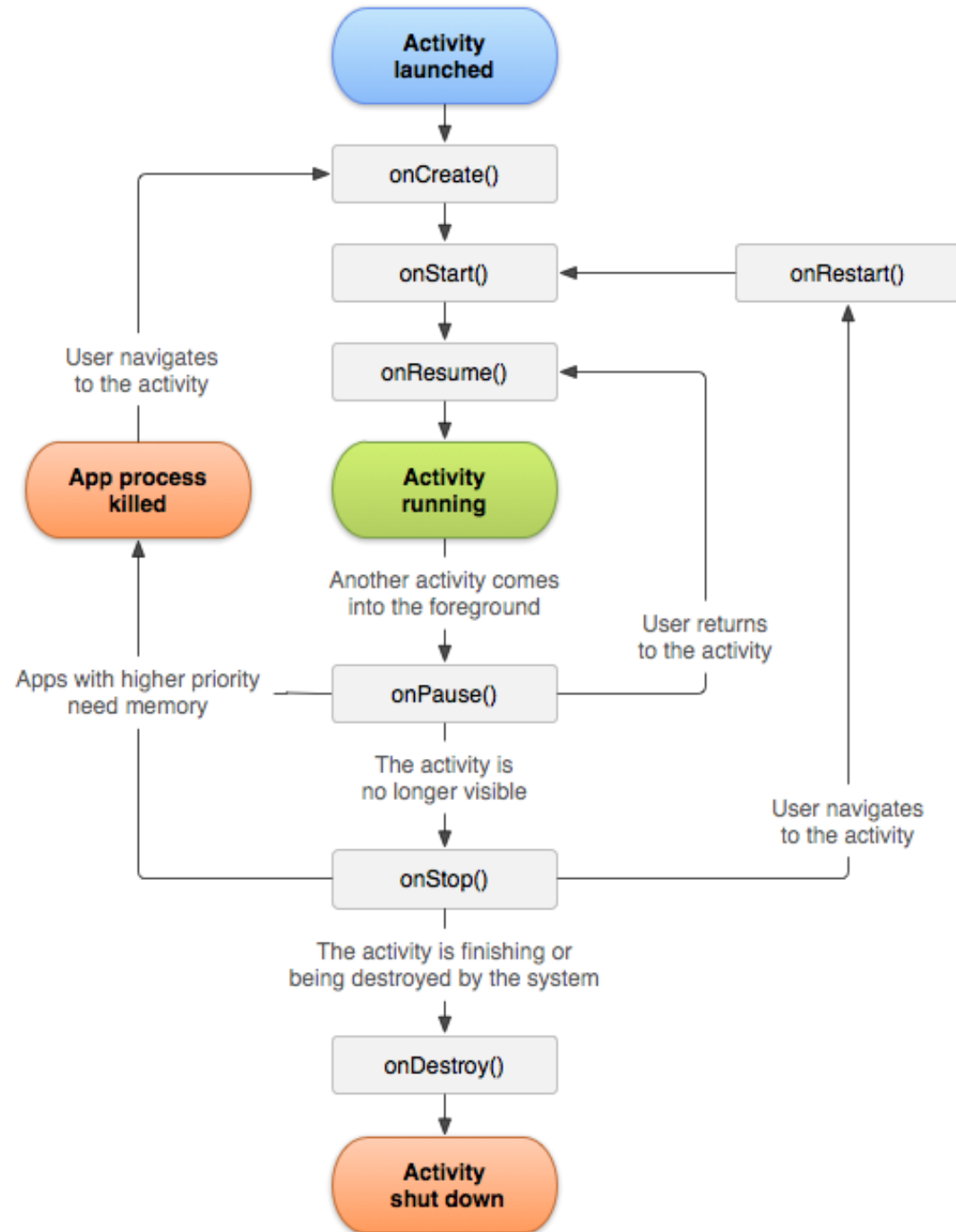
- Κάθε activity έχει το δικό του κύκλο ζωής και είναι υπεύθυνο να αποθηκεύει την δική του κατάσταση κατά το κύκλο ζωής του
- Ο **Activity Manager** παρακολουθεί τις ενεργές activities
 - τις καταστάσεις που βρίσκονται (διαχειρίζεται το κύκλο ζωής των εφαρμογών) και
 - παρέχει την κοινή στοίβα πλοήγησης από εφαρμογή σε εφαρμογή με το back (navigation back stack)



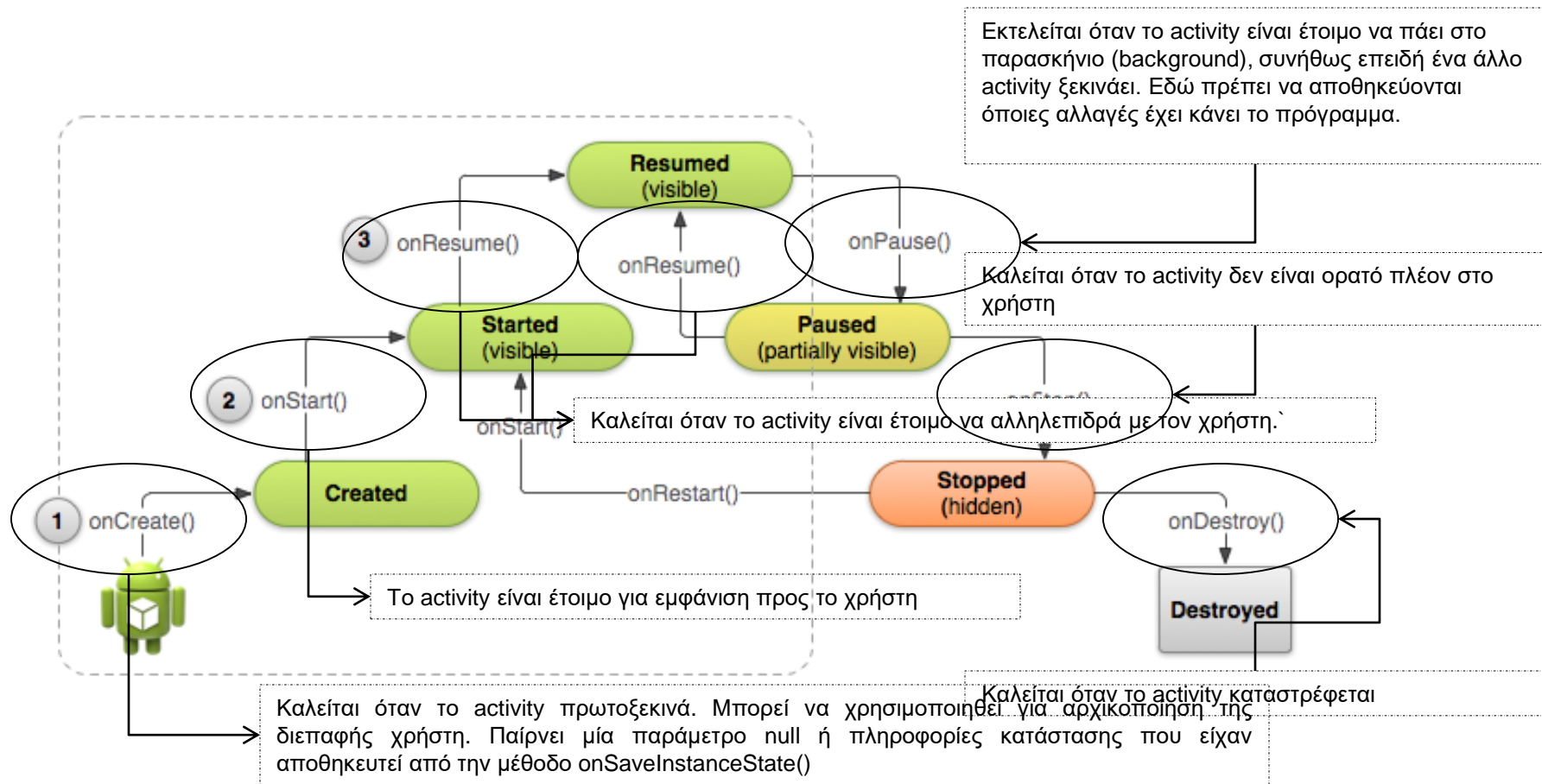
Κλάση Activity

- Κάθε διεπαφή χρήστη (οθόνη) αναπαρίσταται από μία κλάση Activity
- Μία εφαρμογή έχει ένα ή περισσότερα activities και μία διεργασία Linux που τα περιέχει
- **extends Activity** (π.χ. MainActivity extends Activity)
- **override onCreate() callback**
 - Πυροδοτείται όταν το Android σύστημα δημιουργεί την activity
 - Εδώ αρχικοποιεί κανείς τα βασικά συστατικά της activity π.χ. ορίζει το layout της

Κύκλος ζωής ενός Activity



Activity Lifecycle

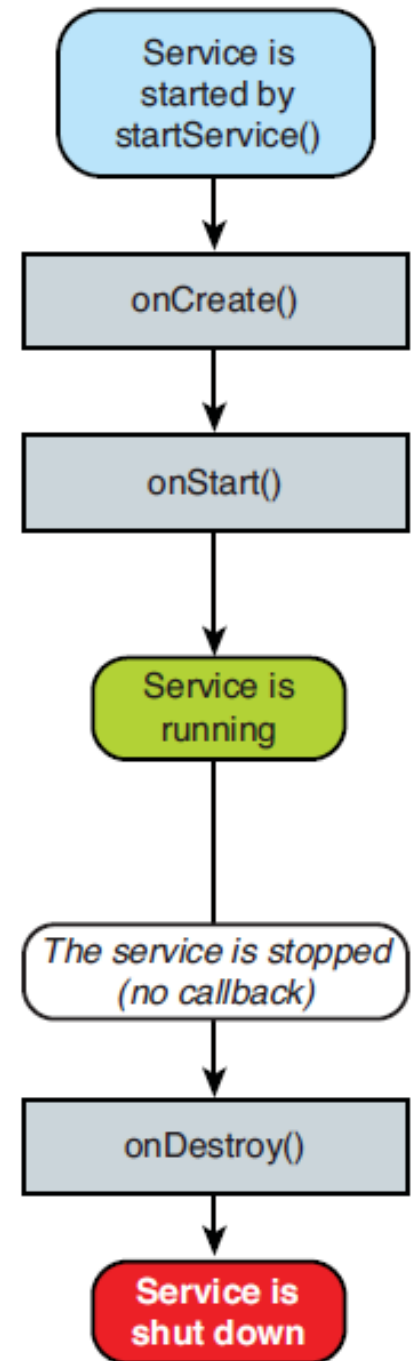


Service ένα συστατικό εφαρμογής

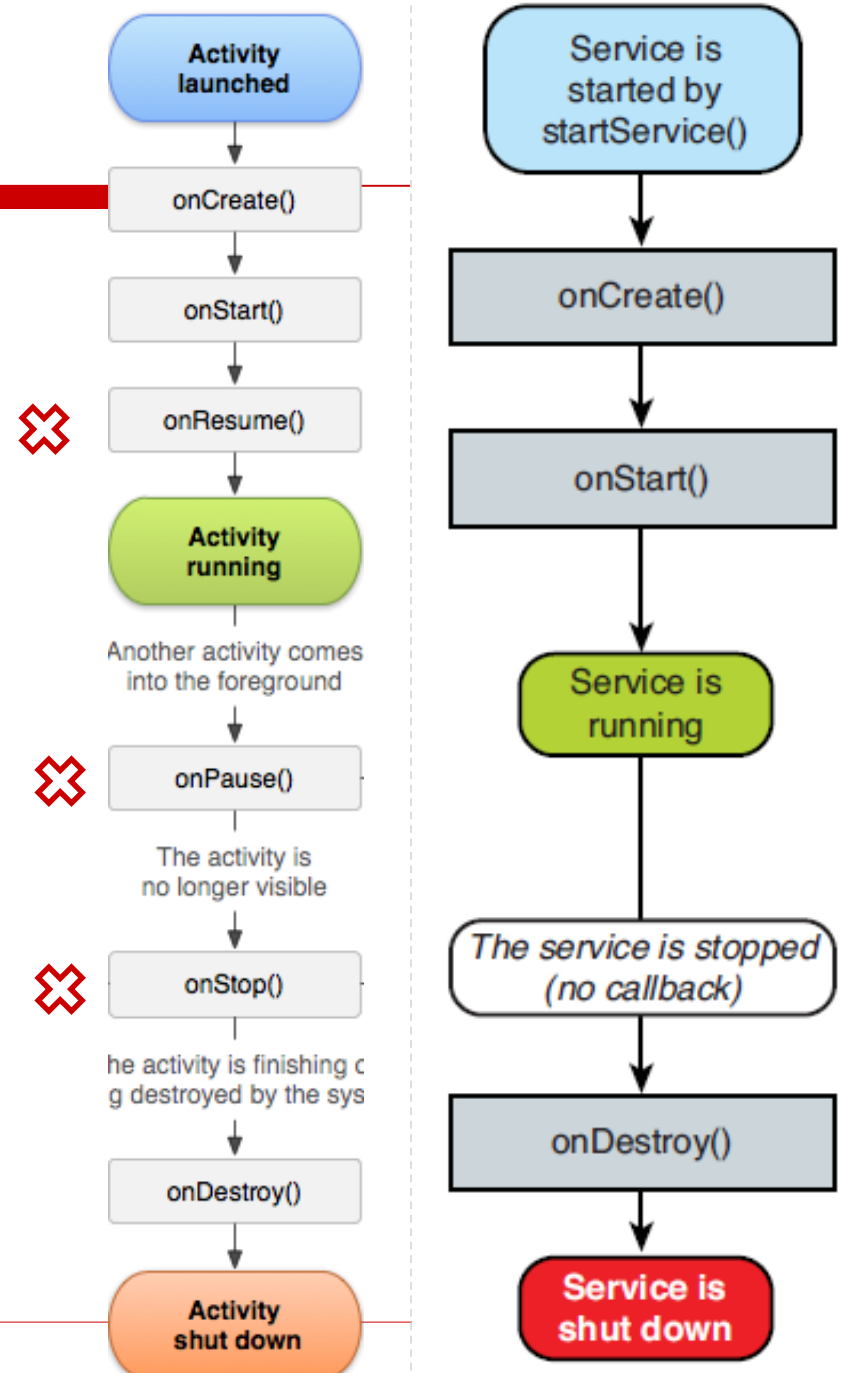
- Διεργασία που τρέχει στο παρασκήνιο χωρίς τη παρέμβαση του χρήστη για να εκτελέσει λειτουργίες
 - λειτουργίες μακράς διάρκειας
 - λειτουργίες για άλλες απομακρυσμένες διεργασίες
- Δεν παρέχει UI στο χρήστη
 - Αναπαραγωγή μουσικής ενώ ο χρήστης είναι σε άλλη app
 - Να κατεβάσει ή συγχρονίσει δεδομένα χωρίς να μπλοκάρει το διάδραση του χρήστη με άλλη app
- Παρόμοιο με Unix daemon
- Υπάρχουν πολλά built-in services σε Android

Η κλάση Service

- Η βασική κλάση για διαδικασίες που εκτελούνται στο παρασκήνιο
 - **extends Service**
 - **override onCreate()**
- Δεν έχει γραφική διεπαφή
- Από default δεν είναι μια ξεχωριστή διεργασία ούτε ένα ξεχωριστό thread.
Αποτελεί τμήμα του main thread.
 - Αν επιθυμούμε μπορούμε να καθορίσουμε ότι θα τρέχει σε ξεχωριστή διεργασία
- Παραδείγματα χρήσης
 - network downloading
 - Αναπαραγωγή μουσικής στο παρασκήνιο
 - TCP/UDP Server



Σύγκριση life cycle



Broadcast receivers

ένα συστατικό εφαρμογής

- Οι ανακοινώσεις broadcast μπορεί να
 - Παράγονται από το λειτουργικό (**OS-generated**) π.χ. χαμηλή στάθμη μπαταρίας, αποκατάσταση σύνδεσης Wifi, πάτημα του πλήκτρου της φωτογραφικής
 - Παράγονται από το χρήστη (**User-generated**) π.χ. έναρξη ή τερματισμός μια διαδικασίας, ενεργοποίηση ενός χαρακτηριστικού
- Η εφαρμογή λαμβάνει ανακοινώσεις broadcast και αντιδρά κατάλληλα
 - Προκαλούν την δημιουργία intents που μπορούν προκαλέσουν την εκτέλεση κώδικα εκτός της κανονικής (ροής regular user flow)
- Το σύστημα μπορεί να στείλει broadcasts ακόμη και σε εφαρμογές που δεν τρέχουν
- Παραδείγματα χρήσης
 - Ενημέρωση άλλων εφαρμογών ότι δεδομένα έχουν κατεβεί
 - Δεν διαθέτουν διεπαφή χρήστη αλλά μπορούν να δημιουργήσουν ένα status bar notification για να ειδοποιήσουν το χρήστη εάν ένα broadcast event συμβαίνει
 - Ως gateway με άλλα συστατικά με σκοπό να κάνει ελάχιστη δουλειά

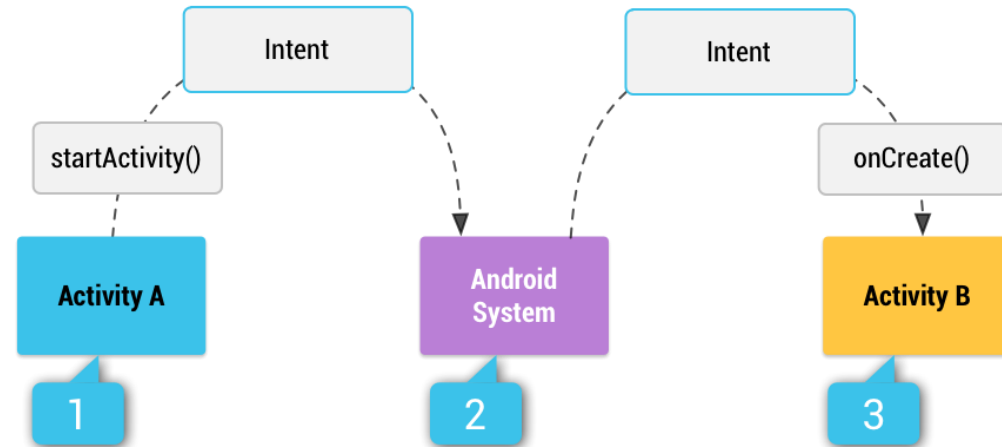
Content Providers

ένα συστατικό εφαρμογής

- Επιτρέπουν σε μια εφαρμογή να έχει πρόσβαση σε δεδομένα άλλων εφαρμογών (π.χ. Contacts)
- Επιτρέπουν σε μια εφαρμογή να διαμοιράζει τα δεδομένα που παράγει σε άλλες εφαρμογές

Εκτέλεση συστατικών: ο μηχανισμός Intent

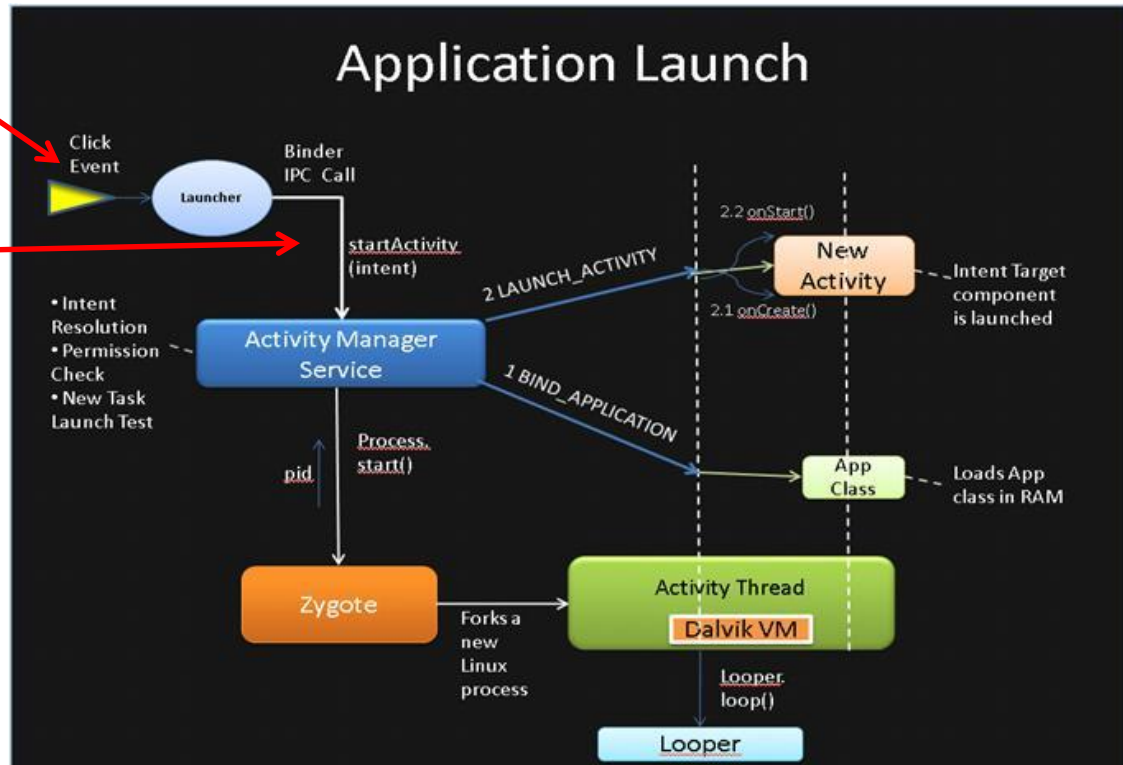
- ❑ Σε android σχεδόν ΟΛΕΣ οι αλληλοεπιδράσεις γίνονται μέσω intents
- ❑ Μηχανισμός που καθορίζει ποια συγκεκριμένη ενέργεια (action) πρέπει να εκτελεστεί στέλνοντας **ασύγχρονα μηνύματα**
- ❑ Ο ρόλος ενός intent είναι να αιτηθεί – στέλνοντας μήνυμα προς τον ActivityManagerService για μια ενέργεια που ο χρήστης επιθυμεί να γίνει
- ❑ Μέσω των intents μπορούμε να ξεκινήσουμε την εκτέλεση των Activities, των Services και των Broadcast Receivers (**όχι Content Providers**)



Intent, έναρξη εφαρμογής

Πάτημα κουμπιού εφαρμογής

Ειδοποίηση του ActivityManager σχετικά με την ενέργεια που επιθυμεί ο χρήστης



ΤΑ ΠΡΩΤΑ ΣΑΣ ΒΗΜΑΤΑ ΣΤΟ ANDROID SDK

Εγκατάσταση Android SDK

- Λήψη του Android SDK
- Εγκατάσταση plugin ανάλογα με το περιβάλλον IDE
 - **Android Studio**
 - ADT plugin για Eclipse
 - NBAndroid plugin για Netbeans
- Λήψη SDK tools and platforms μέσω του SDK Manager

Διάθεση Android SDK

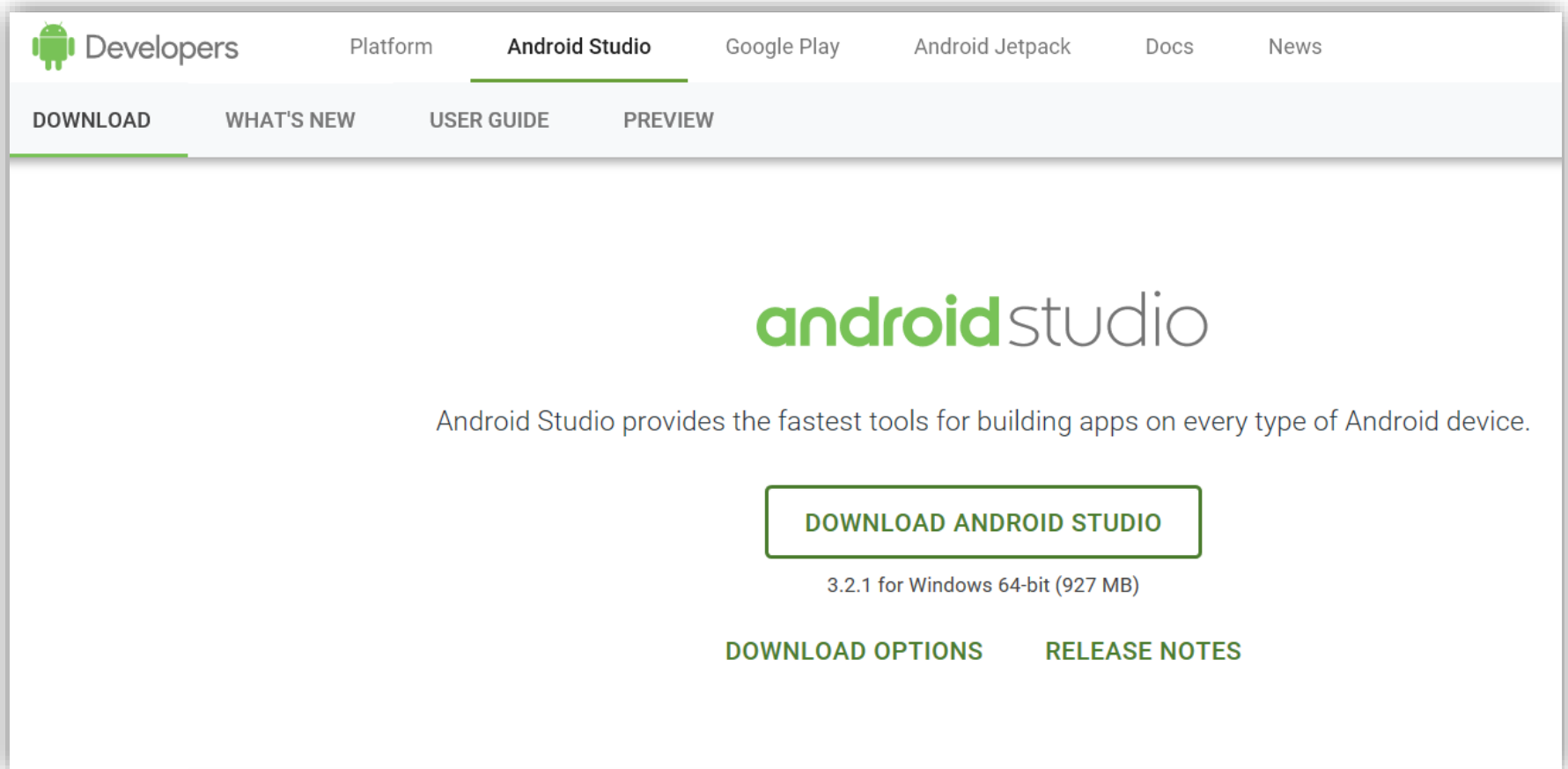
<https://developer.android.com/studio>

User Guides Android SDK

<https://developer.android.com/studio/intro>

<https://developer.android.com/guide/>

Διάθεση Android SDK



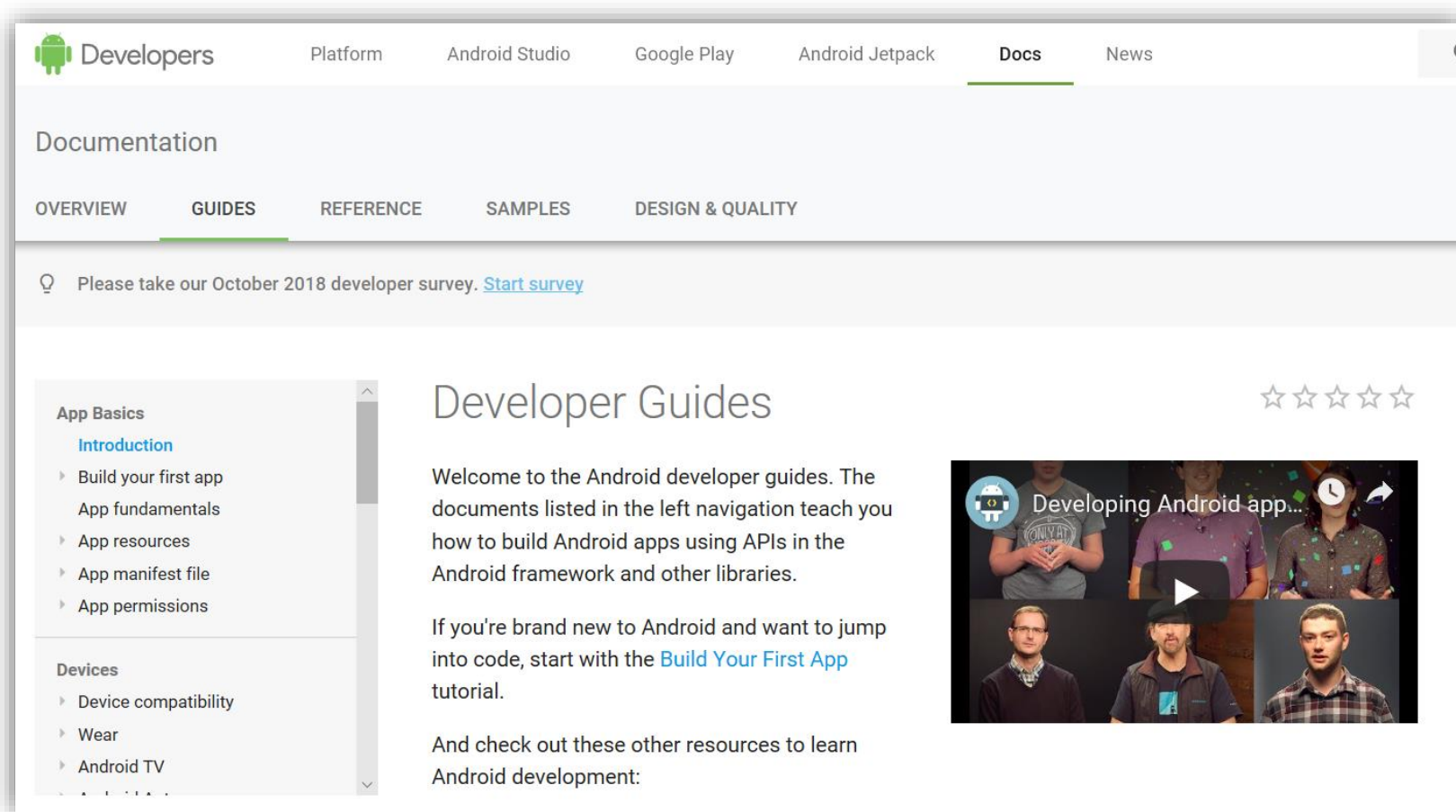
The screenshot shows the 'Android Studio' page on the 'Android Developers' website. The navigation bar includes 'Platform', 'Android Studio' (highlighted), 'Google Play', 'Android Jetpack', 'Docs', and 'News'. Below the navigation bar are links for 'DOWNLOAD', 'WHAT'S NEW', 'USER GUIDE', and 'PREVIEW'. The main content area features the 'androidstudio' logo, a description: 'Android Studio provides the fastest tools for building apps on every type of Android device.', a prominent 'DOWNLOAD ANDROID STUDIO' button, and the version information '3.2.1 for Windows 64-bit (927 MB)'. At the bottom of the content area are links for 'DOWNLOAD OPTIONS' and 'RELEASE NOTES'.

Σύνδεσμοι:

<https://developer.android.com/studio>

<https://developer.android.com/studio/intro/>

Οδηγίες για την πρώτη σας εφαρμογή



The screenshot shows the Android Developers website. At the top, there are navigation links for Platform, Android Studio, Google Play, Android Jetpack, Docs (which is highlighted), and News. Below this is the 'Documentation' section with sub-links for OVERVIEW, GUIDES (highlighted), REFERENCE, SAMPLES, and DESIGN & QUALITY. A notification banner asks to take the October 2018 developer survey. The main content area is titled 'Developer Guides' and includes a five-star rating. The text welcomes users to the guides and provides instructions on how to get started, including a link to the 'Build Your First App' tutorial. A video thumbnail titled 'Developing Android app...' is also visible. On the left, a sidebar menu lists 'App Basics' (Introduction, Build your first app, App fundamentals, App resources, App manifest file, App permissions) and 'Devices' (Device compatibility, Wear, Android TV).

Σύνδεσμος: <https://developer.android.com/guide/>
<https://developer.android.com/training/basics/firstapp/index.html>

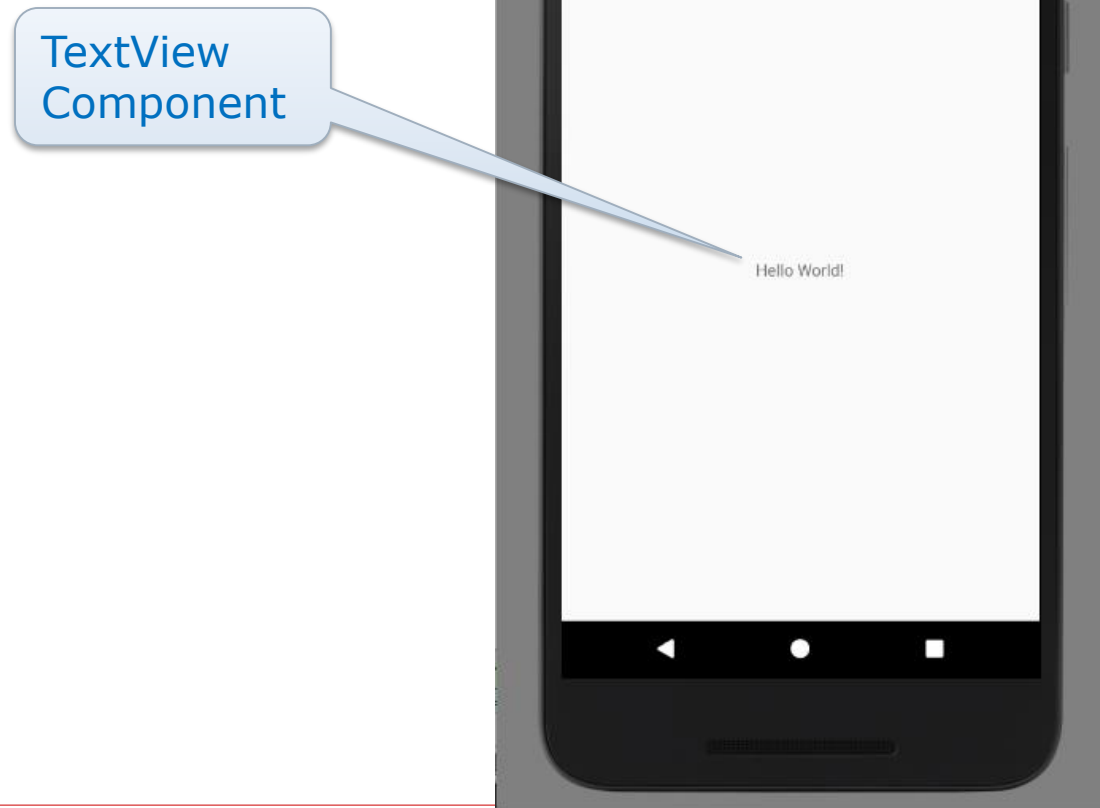
Εκτέλεση εφαρμογής

- ❑ Emulator
- ❑ Στην Android συσκευή σας



Παράδειγμα εφαρμογής με ένα Activity

□ Hello World !



Δομή αρχείων σε Android Projects

Source code

Auto-generated code

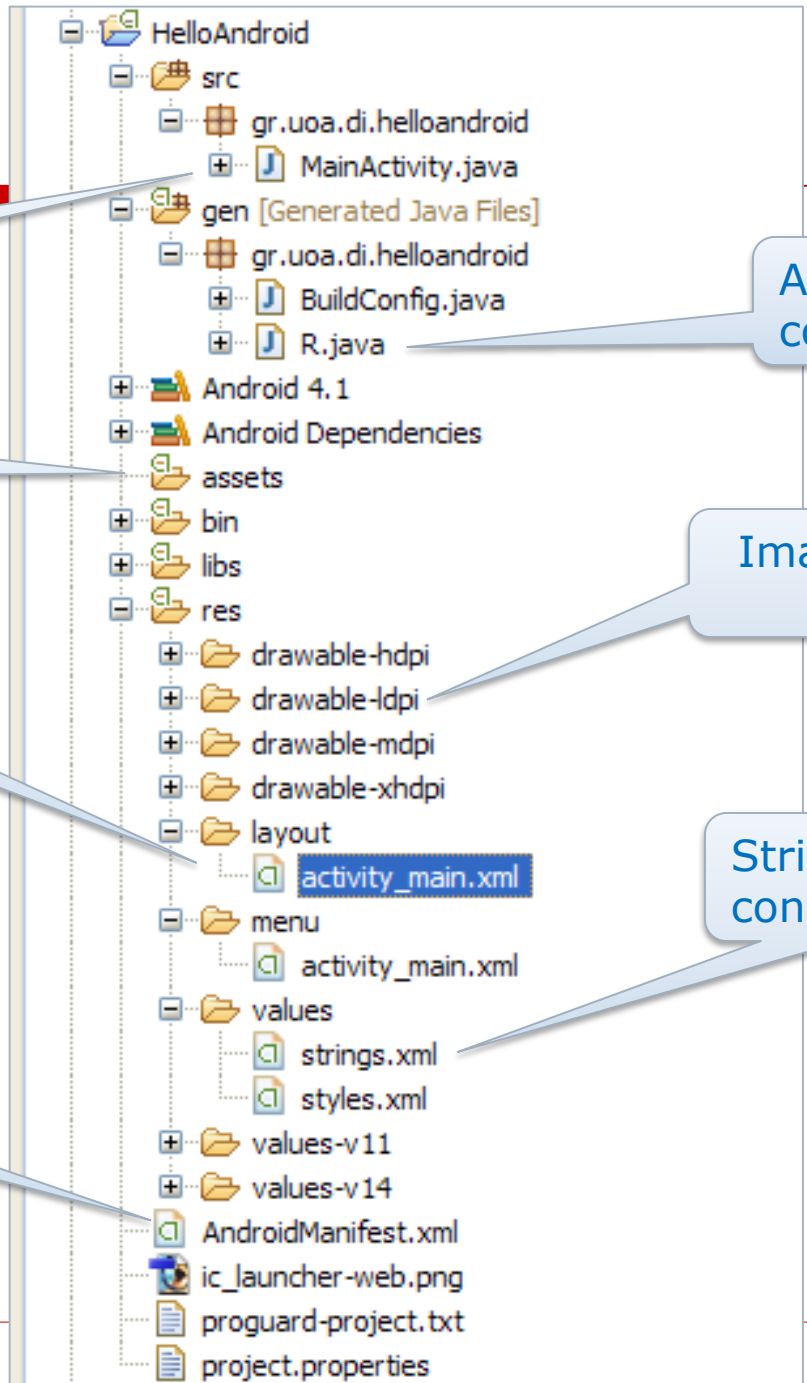
Files

Images

UI layout

String constants

Configuration



Android manifest file

Κάθε εφαρμογή έχει ένα αρχείο AndroidManifest.xml που παρέχει σημαντικές πληροφορίες για την εφαρμογή στο OS

- ❑ Δίνει όνομα στο Java πακέτο για την εφαρμογή το οποίο αποτελεί μοναδικό αναγνωριστικό για την εφαρμογή.
- ❑ Καθορίζει το ελάχιστο επίπεδο Android API που απαιτείται από την εφαρμογή.
- ❑ Περιέχει λίστα με τις βιβλιοθήκες με τις οποίες πρέπει να συνδεθεί η εφαρμογή.
- ❑ Καθορίζει **τα δικαιώματα** που πρέπει να έχει η εφαρμογή
 - για να έχει πρόσβαση σε προστατευόμενα κομμάτια του API
 - για να αλληλεπιδρά με άλλες εφαρμογές
 - τα δικαιώματα που πρέπει να έχουν άλλες εφαρμογές για να αλληλεπιδρούν με αυτή
- ❑ Δηλώνει τα χαρακτηριστικά (h/w & s/w), π.χ. κάμερα, Bluetooth κ.α., που χρησιμοποιεί η εφαρμογή.
- ❑ **Περιγράφει τα components της εφαρμογής (activities, services, content providers κλπ.).**

```
<?xml version="1.0" encoding="utf-8"?>

<manifest>

    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <instrumentation />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />
    <compatible-screens />
    <supports-gl-texture />

    <application>

        <activity>
            <intent-filter>
                <action />
                <category />
                <data />
            </intent-filter>
            <meta-data />
        </activity>

        <activity-alias>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </activity-alias>

        <service>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </service>

        <receiver>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </receiver>

        <provider>
            <grant-uri-permission />
            <meta-data />
            <path-permission />
        </provider>

        <uses-library />

    </application>

</manifest>
```

Android manifest file (2/2)

□ Παράδειγμα από την "HelloWorld" app

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="gr.uoa.di.helloandroid"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

SDK versions

Application permissions

Activities

Intents

Απλό αρχείο string resources

/res/values/strings.xml

String Hello word!

```
<resources>
  <string name="app_name">HelloAndroid2</string>
  <string name="hello_world">Hello world!</string>
  <string name="menu_settings">Settings</string>
  <string name="title_activity_main">MainActivity</string>
</resources>
```

Κλάση Activity

- extends Activity (π.χ. MainActivity extends Activity)
- override onCreate() callback
 - Πυροδοτείται όταν το Android σύστημα δημιουργεί την activity
 - Εδώ αρχικοποιεί κανείς τα βασικά συστατικά της activity π.χ. ορίζει το layout της

Απλή Κλάση Activity

/src/gr/uoa/di/helloandroid/MainActivity.java

```
package gr.uoa.di.helloandroid;

import android.os.Bundle;
import android.app.Activity;

public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Τρέχει ότι ορίζεται
στην onCreate()
της γονικής Activity

Κώδικας για
επιπλέον
λειτουργικότητα
από της γονικής
onCreate()

UI Layout
activity_main.xml

MainActivity

MyApplication - [C:\Users\pantelis\AndroidStudioProjects\MyApplication] - [app] - ...\app\src\main\java\com\example\pantelis\myapplication\MainActivity.java

The screenshot displays the Android Studio IDE with the following components:

- Menu Bar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help.
- Toolbar:** Standard IDE navigation and development icons.
- Breadcrumb:** MyApplication > app > src > main > java > com > example > pantelis > myapplication.
- Project View (Left):** Shows the project structure with folders like manifests, java, res, and Gradle Scripts. The package com.example.pantelis.myapplication is expanded, showing MainActivity.java.
- Code Editor (Right):** Displays the following Java code:

```
1 package com.example.pantelis.myapplication;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

Απλό UI Layout XML

/res/layout/activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/hello_world"
        tools:context=".MainActivity" />

</RelativeLayout>
```

TextView
Component

String hello_word

Layout xml file

MyApplication - [C:\Users\pantelis\AndroidStudioProjects\MyApplication] - [app] - ...app\src\main\res\layout\activity_main.xml - Android Studio 2.3.3

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

MyApplication app src main res layout activity_main.xml

Android activity_main.xml x MainActivity.java x

Project Structure

- app
 - manifests
 - AndroidManifest.xml
 - java
 - com.example.pantelis.myapplication
 - MainActivity
 - com.example.pantelis.myapplication (androidTest)
 - com.example.pantelis.myapplication (test)
 - res
 - drawable
 - layout
 - activity_main.xml
 - mipmap
 - values
 - colors.xml
 - strings.xml
 - styles.xml
- Gradle Scripts

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.example.pantelis.myapplication.MainActivity">
8
9     <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Hello World!"
13         app:layout_constraintBottom_toBottomOf="parent"
14         app:layout_constraintLeft_toLeftOf="parent"
15         app:layout_constraintRight_toRightOf="parent"
16         app:layout_constraintTop_toTopOf="parent" />
17
18 </android.support.constraint.ConstraintLayout>
19
```

strings.xml

MyApplication - [C:\Users\pantelis\AndroidStudioProjects\MyApplication] - [app] - ...app\src\main\res\values\strings.xml - Android Studio 2.3.3

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

MyApplication > app > src > main > res > values > strings.xml

1: Project
Android
app
 manifests
 AndroidManifest.xml
 java
 com.example.pantelis.myapplication
 MainActivity
 com.example.pantelis.myapplication (androidTest)
 com.example.pantelis.myapplication (test)
 res
 drawable
 layout
 activity_main.xml
 mipmap
 values
 colors.xml
 strings.xml
 styles.xml

activity_main.xml x strings.xml x MainActivity.java x

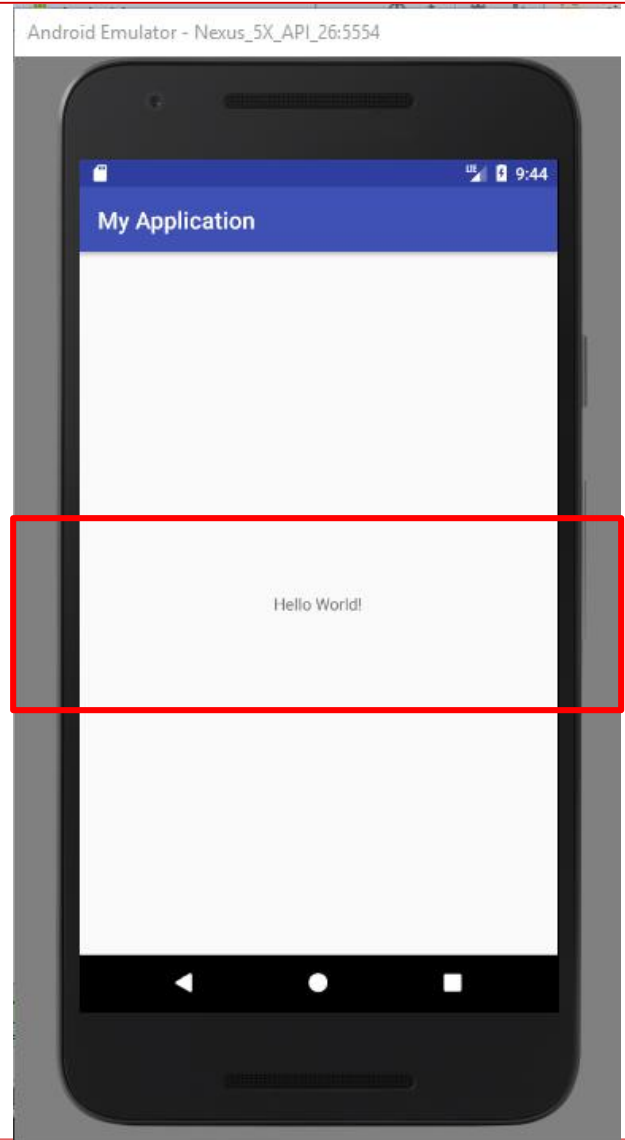
Edit translations for all locales in the translations editor.

```
resources
1 <resources>
2   <string name="app_name">My Application</string>
3 </resources>
4
```

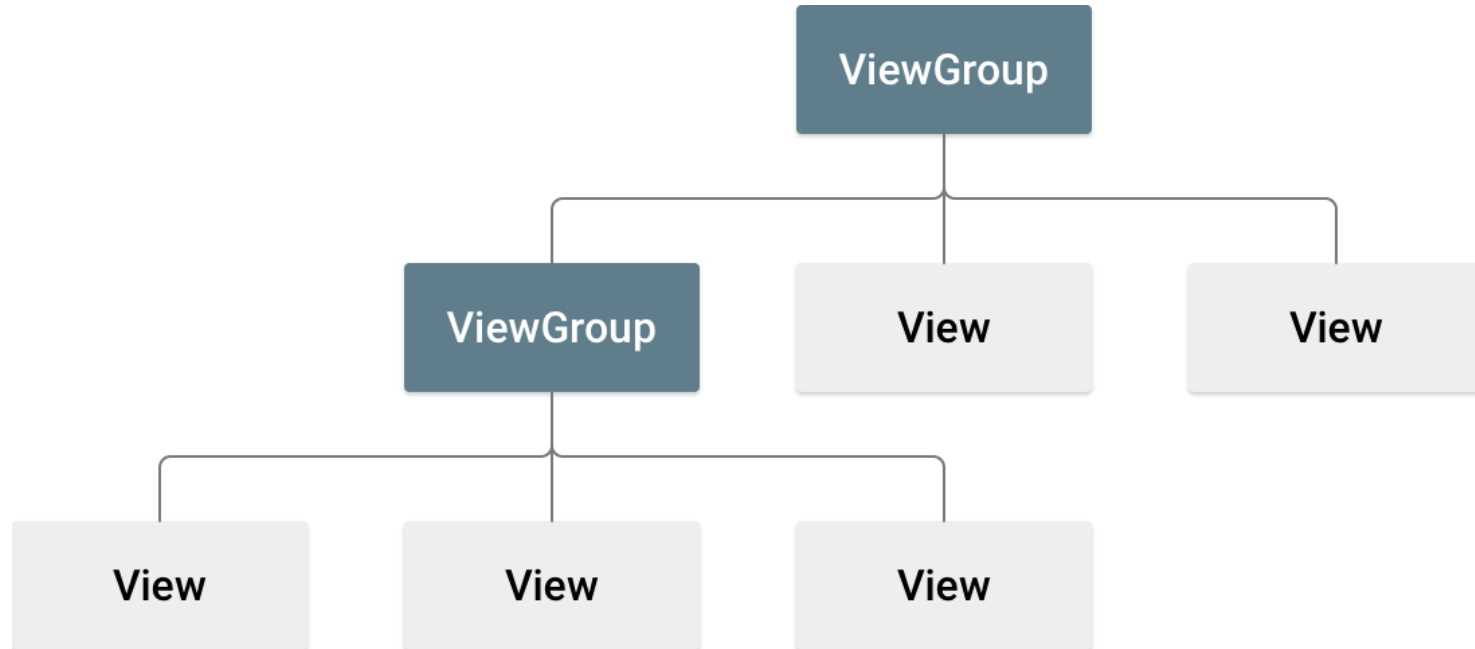
Emulator

TextView
Component

Hello World!



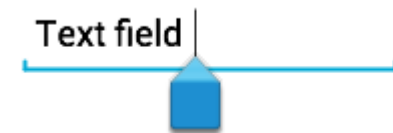
User Interface Layout



- ❑ Ότι εμφανίζετε σε μία οθόνη είναι view
- ❑ Ένα view μπορεί να είναι ένα **input control** ή **widgets** που συνθέτει κάποιο τμήμα του UI.
- ❑ Τα views ομαδοποιούνται σε ViewGroup. Κάθε view group είναι ένα αόρατο container που οργανώνει views.
- ❑ Το UI Layout ορίζεται ως μία ιεραρχία από ViewGroups και View αντικείμενα

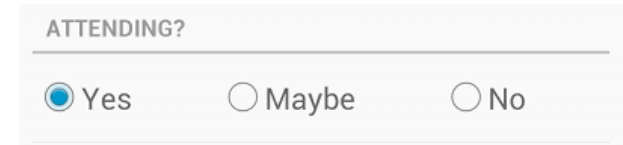
Input controls (1)

Control Type	Description	Related Classes
Button	A push-button that can be pressed , or clicked , by the user to perform an action.	Button
Text field	An editable text field . You can use the <code>AutoCompleteTextView</code> widget to create a text entry widget that provides auto-complete suggestions	EditText , AutoCompleteTextView
Checkbox	An on/off switch that can be toggled by the user. You should use checkboxes when presenting users with a group of selectable options that are not mutually exclusive .	CheckBox



Input controls (2)

Control Type	Description	Related Classes
Radio button	Similar to checkboxes, except that only one option can be selected in the group .	RadioGroup RadioButton
Toggle button	An on/off button with a light indicator.	ToggleButton
Spinner	A drop-down list that allows users to select one value from a set.	Spinner
Pickers	A dialog for users to select a single value for a set by using up/down buttons or via a swipe gesture. Use a <code>DatePicker</code> widget to enter the values for the date (month, day, year) or a <code>TimePicker</code> widget to enter the values for a time (hour, minute, AM/PM), which will be formatted automatically for the user's locale.	DatePicker , TimePicker



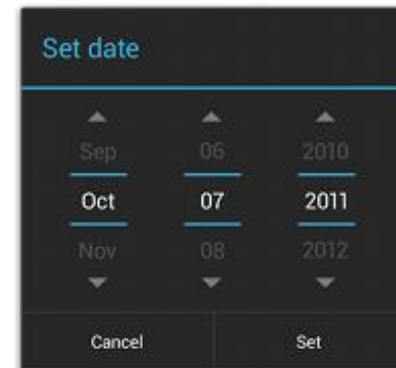
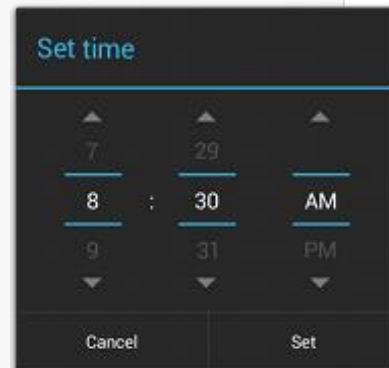
Home

Home

Work

Other

Custom



Layouts: View Groups (1)



Linear Layout: Οργανώνει τα views σε μονές οριζόντιες ή κάθετες γραμμές

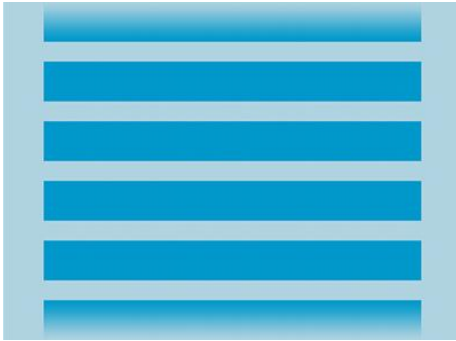


Relative Layout: Οργανώνει τα child-views σε σημεία που σχετίζονται με άλλα στοιχεία και σχετικά με άλλα child-views (child A αριστερά από το child B) ή στη πάνω περιοχή του γονέα (parent)

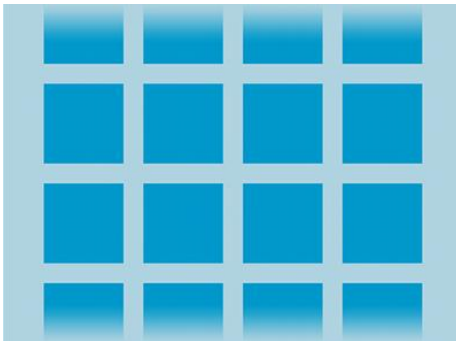


Web View: Απεικονίζει web σελίδες

Layouts: View Groups (2)



List View : Απεικονίζει μία **κυλιόμενη** λίστα μονών γραμμών

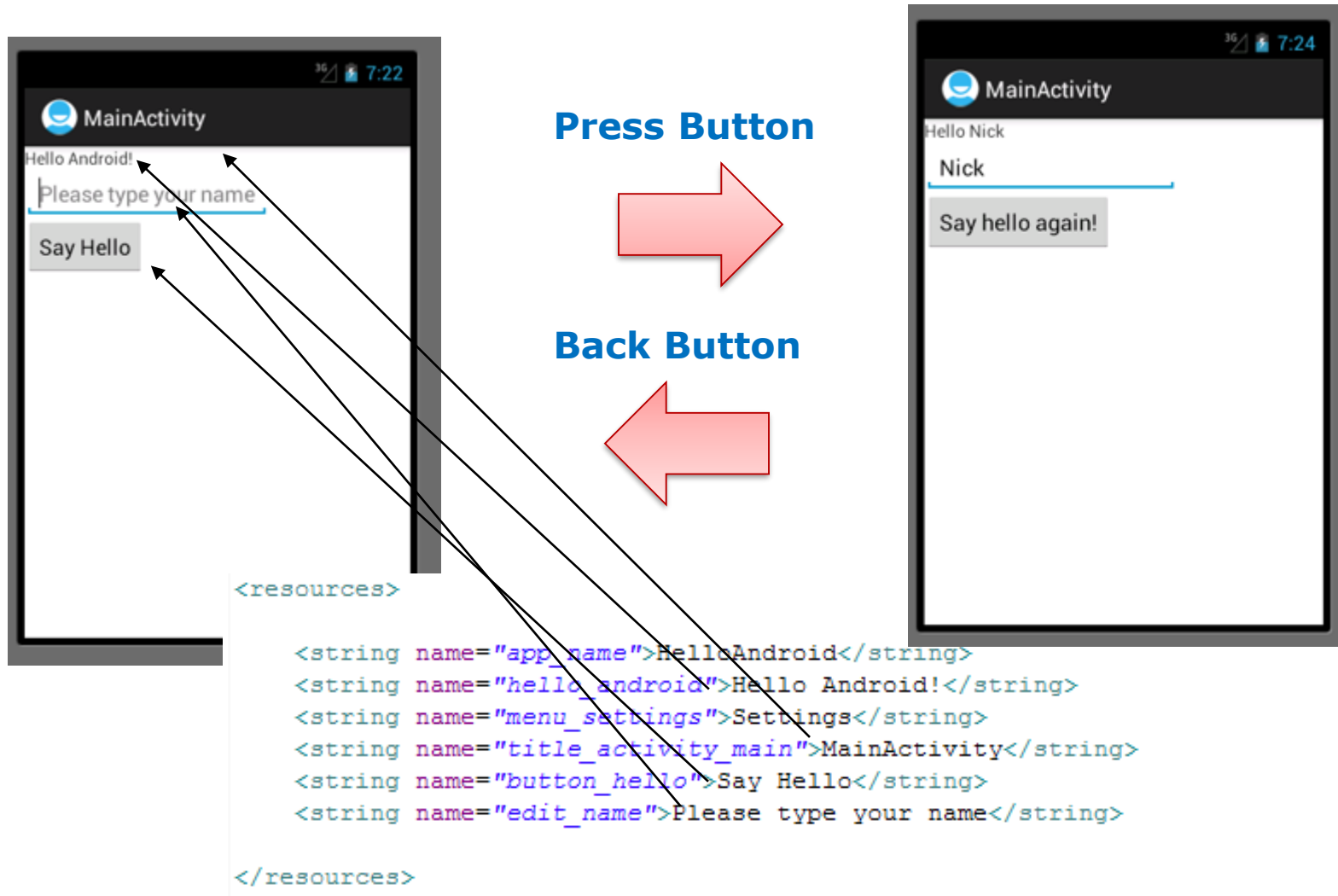


Grid View: Απεικονίζει ένα δισδιάστατο **κυλιόμενο** πλέγμα

Inputs control @ layout xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal">
    <EditText android:id="@+id/edit_message"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />
    <Button android:id="@+id/button_send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send"
        android:onClick="sendMessage" />
</LinearLayout>
```

Παράδειγμα με 3 Control Inputs



UI Layout XML (Παράδειγμα)

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView android:id="@+id/hello_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_android"
        tools:context=".MainActivity" />
    <EditText android:id="@+id/edit_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="@string/edit_name" />
    <Button
        android:id="@+id/button_hello"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_hello"
        android:onClick="sayHello" />
</LinearLayout>
```

TextView
Component

EditText
Component

Κουμπι button_hello

String button_hello

Όνομα μεθόδου
που θα κληθεί όταν
πατηθεί το κουμπι

Κώδικας

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.EditText;
import android.app.Activity;
```

```
public class MainActivity extends Activity {
```

```
private Button helloButton;
private EditText editTextName;
private TextView textViewHello;
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

```
helloButton = (Button) findViewById(R.id.button_hello);
editTextName = (EditText) findViewById(R.id.edit_name);
textViewHello = (TextView) findViewById(R.id.hello_message);
```

```
}
```

```
public void sayHello(View view) {
    textViewHello.setText("Hello " + editTextName.getText());
    helloButton.setText("Say hello again!");
```

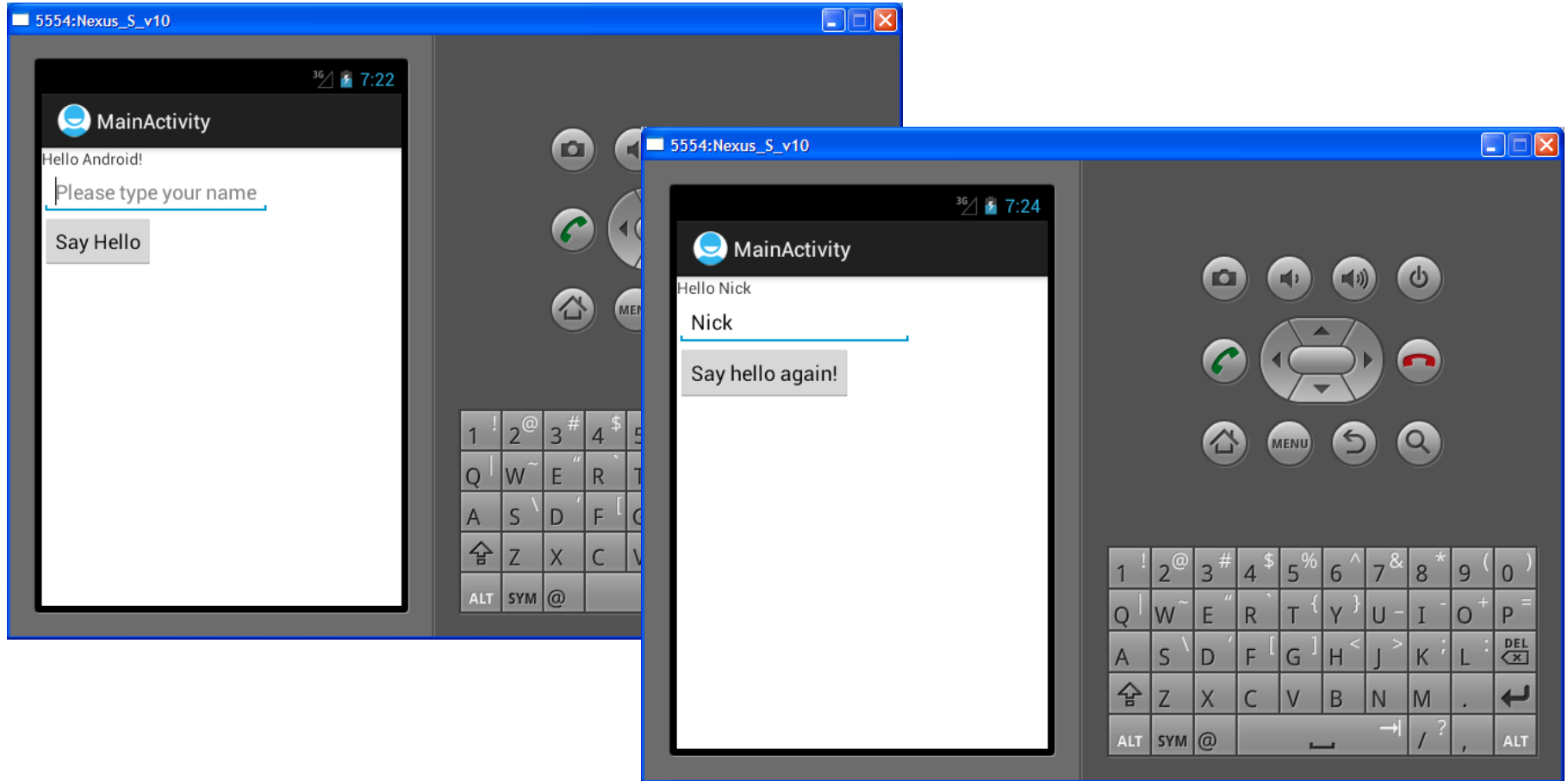
```
}
```

UI Layout
activity_main.xml

References to
components of
activity_main.xml

Μέθοδος που
καλείται όταν
πατηθεί το κουμπί

Εκτέλεση στον Emulator (Παράδειγμα)



Άλλες αναφορές για το σχεδιασμό UI

- <https://developer.android.com/design/index.html>
- <https://developer.android.com/training/basics/firstapp/building-ui.html>

Πως χρησιμοποιούμε το μηχανισμό Intent

ΠΑΡΑΔΕΙΓΜΑ ΜΕ 2 ACTIVITIES

Η κλάση Intent (1)

- Ο ρόλος ενός intent είναι να ειδοποιηθεί το `ActivityManagerService` ότι ο χρήστης επιθυμεί να γίνει κάποια ενέργεια.
- Ένα intent είναι ένα αντικείμενο της κλάσης `Intent` το οποίο περιέχει κάποιο περιεχόμενο (το μήνυμα του intent)
- Μέσω των intents μπορούμε να ξεκινήσουμε την εκτέλεση
 - των `Activities`,
 - των `Services` και
 - των `Broadcast Receivers`

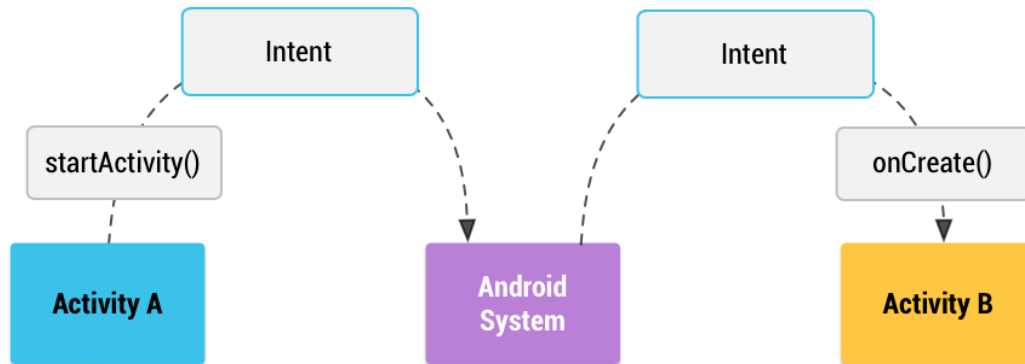
Η κλάση Intent (2)

- Ένα **service** μπορεί να κληθεί με χρήση της
 - `Context.startService(Intent service)`
- Ένα **activity** μπορεί να κληθεί με χρήση των
 - `Context.startActivity(Intent intent)`
 - `Activity.startActivityForResult(Intent intent, int requestCode)`
- Μια εφαρμογή μπορεί να δημιουργήσει ένα **broadcast μήνυμα** με χρήση του Intent
 - `Context.sendBroadcast(Intent intent),`
 - `Context.sendOrderedBroadcast(Intent intent, String receiverPermission)`
 - `Context.sendStickyBroadcast(Intent intent)`

Η κλάση IntentService (Service)

- Χρησιμοποιείται
 - Για διαδικασίες που απαιτούν πολύ χρόνο.
 - Όταν απαιτείται callback (intent)
- Καλείται μέσω intent μόνο από το main Thread
 - Συνήθως δεν απαιτείται επικοινωνία με το main thread
 - Αν χρειαστεί επικοινωνία με το main thread, γίνεται με τη βοήθεια handler ή broadcast intents
- Εκτελείται σε ξεχωριστό νήμα (worker thread)
- ΔΕΝ μπορεί να εκτελέσει πράγματα παράλληλα
- Πολλαπλά intents δημιουργούν ουρά εκτελέσεων στο ίδιο worker thread

Intent activities



1. Η activity A “MainActivity” δημιουργεί ένα Intent μήνυμα με την περιγραφή της ενέργειας και την στέλνει στο Android σύστημα με τη `startActivity()`
2. Το Android σύστημα καλεί την `onCreate()` της activity’s B “HelloActivity” και περνά το Intent μήνυμα
3. Η “HelloActivity” έρχεται στο προσκήνιο κ.τ.λ.

Παράδειγμα με 2 Activities

Χρήση Intent για έναρξη νέας activity

1^η Activity: MainActivity

```
public void sayHello(View view) {  
  
    Intent intent = new Intent(this, HelloActivity.class);  
    String name = editTextName.getText().toString();  
    intent.putExtra("name", name);  
    startActivity(intent);  
}
```

Δημιουργία μηνύματος Intent με την νέα activity

Προσθήκη δεδομένων στο Intent

2^η Activity: HelloActivity

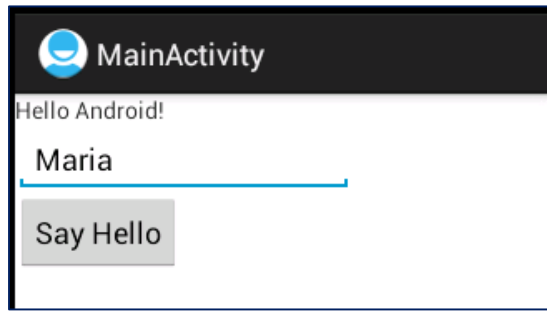
```
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    Intent intent = getIntent();  
    String name = intent.getStringExtra("name");  
    TextView helloTextView = new TextView(this);  
    helloTextView.setText("Hello " + name);  
    setContentView(helloTextView);  
}
```

Activity Manager

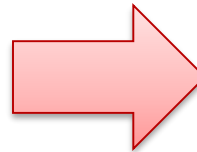
Προσπέλαση μηνύματος Intent

Προσπέλαση δεδομένων από το Intent

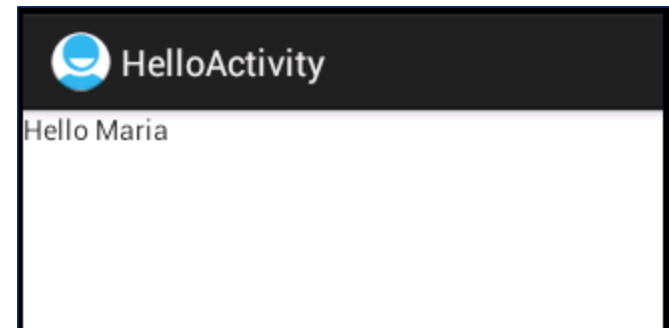
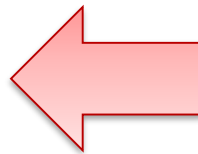
Εκτέλεση στον Emulator (Intent)



Press Button

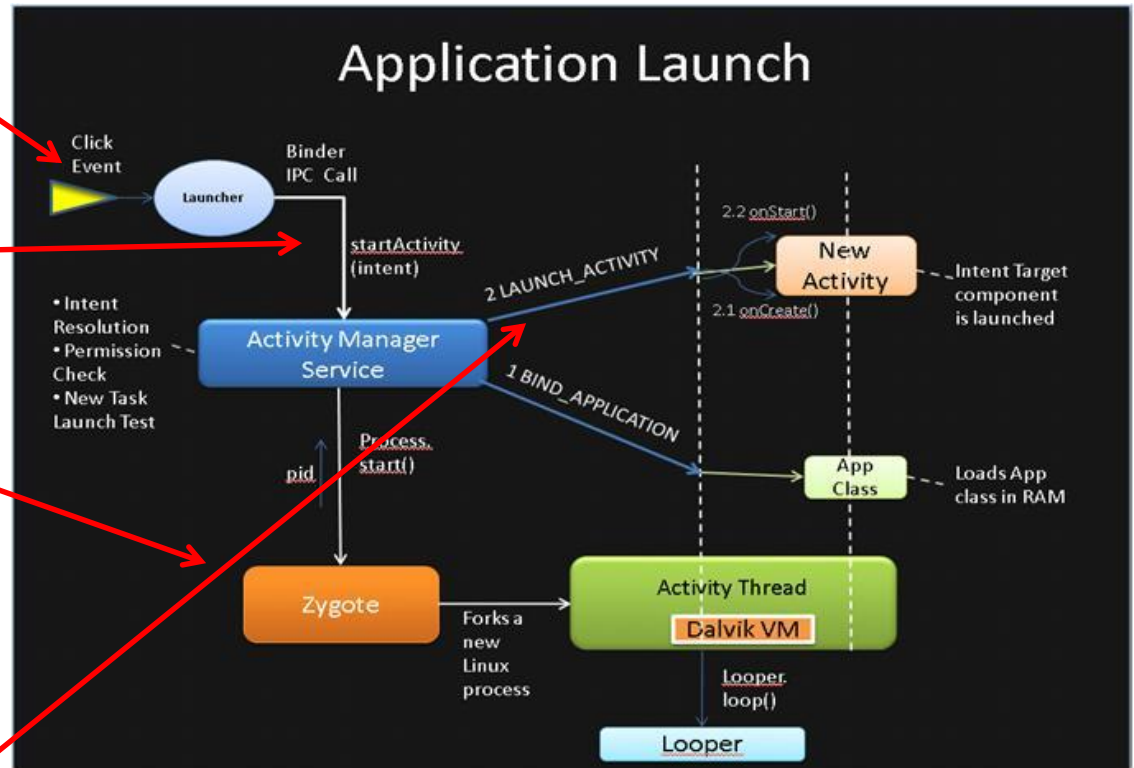


Back Button



Παράδειγμα Intent – Εκκίνηση Εφαρμογής¹

- Πάτημα κουμπιού εφαρμογής
- Ειδοποίηση του ActivityManager σχετικά με την ενέργεια που επιθυμεί ο χρήστης
- Έλεγχος και δημιουργία Διεργασίας για την εφαρμογή που εκκίνησε ο χρήστης
- Δημιουργία του Activity που έχει δηλωθεί ως αρχικό στο Android Manifest xml αρχείο της εφαρμογής



1 <http://coltf.blogspot.gr/p/android-os-processes-and-zygote.html>

Η κλάση Pending Intent

- Η κλάση Pending Intent δίνει τη δυνατότητα εκτέλεσης (επιτρέπει, παρέχει τη δικαιοδοσία) μίας λειτουργίας από άλλη εφαρμογή όπως
 - NotificationManager
 - AlarmManager
 - HomeScreen AppWidgetManager
- Παρέχοντας ένα PendingIntent σε μια άλλη εφαρμογή B, της παρέχεις το δικαίωμα να εκτελέσει τη λειτουργία που προδιαγράφεις με τα δικαιώματα και ταυτότητα της δικής σου εφαρμογής A
- Αντιθέτα, στο Intent όπου η λειτουργία α εκτελεστεί με τα δικαιώματα της άλλης εφαρμογής B

ΆΛΛΕΣ ΣΗΜΑΝΤΙΚΕΣ ΚΛΑΣΕΙΣ

Η κλάση BroadcastReceiver

- Λαμβάνει ανακοινώσεις broadcast και αντιδρά κατάλληλα
 - **extends BroadcastReceiver**
 - implements **onReceive()**: καλείται όταν ο BroadcastReceiver λαμβάνει ένα Intent broadcast
- Οι ανακοινώσεις broadcast μπορεί να
 - Παράγονται από το λειτουργικό (OS-generated) π.χ. χαμηλή στάθμη μπαταρίας, αποκατάσταση σύνδεσης Wifi, πάτημα του πλήκτρου της φωτογραφικής
 - Παράγονται από το χρήστη (User-generated) π.χ. έναρξη ή τερματισμός μια διαδικασίας, ενεργοποίηση ενός χαρακτηριστικού

System Events

Event	Description
Intent.ACTION_BOOT_COMPLETED	To boot έγινε με επιτυχία .Απαιτείται η άδεια, android.permission.RECEIVE_BOOT_COMPLETED.
Intent.ACTION_POWER_CONNECTED	Έγινε σύνδεση του φορτιστή στην συσκευή.
Intent.ACTION_POWER_DISCONNECTED	Έγινε αποσύνδεση του φορτιστή από την συσκευή.
Intent.ACTION_BATTERY_LOW	Η στάθμη της μπαταρίας είναι χαμηλή. Χρησιμοποιείται για την εξοικονόμηση ενέργειας από εφαρμογές που καταναλώνουν πολύ.
Intent.ACTION_BATTERY_OKAY	Η στάθμη της μπαταρίας είναι σε καλό επίπεδο.

BroadcastReceiver – Παράδειγμα

Παράδειγμα χρήσης BroadcastReceiver για ανάκτηση πληροφοριών σχετικά με το Wifi

Manifest File

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>  
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
```


BroadcastReceiver – Παράδειγμα

WifiReceiver

```
package gr.uoa.di.android.monitoring.application;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.net.wifi.WifiInfo;
import android.net.wifi.WifiManager;
import android.os.Bundle;

public class WifiReceiver extends BroadcastReceiver{

    @Override
    public void onReceive(Context context, Intent intent) {

        Bundle b = intent.getExtras();
        WifiInfo info = (WifiInfo) b.get(WifiManager.EXTRA_WIFI_INFO);

        String ssid = info.getSSID();
        String bssid = info.getBSSID();
    }
}
```

BroadcastReceiver – Παράδειγμα

Receiver

```
WifiManager wifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);

WifiReceiver wifiReceiver = new WifiReceiver();
registerReceiver(wifiReceiver, new IntentFilter(
    WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));

wifiManager.startScan();
```

WifiReceiver extends
BroadcastReceiver

Register a BroadcastReceiver to listen to SCAN_RESULTS_AVAILABLE_ACTION. You can either dynamically register an instance of this class with `Context.registerReceiver()` or statically declare an implementation with the `<receiver>` tag in your `AndroidManifest.xml`.

Η κλάση Log

- Η εκτύπωση στη standard έξοδο (System.out.println) ΔΕΝ δουλεύει
- Χρήση της κλάσης Log για debugging
 - Ενεργοποίηση του LogCat (στο Android Studio):
Window-> Show View -> Other -> Android-> Logcat
 - Η κλάση Log παρέχει αρκετές static μεθόδους για εκτύπωση μηνυμάτων διαφορετικής σημασίας για το λειτουργικό σύστημα:

Log.e(): Errors Log.w(): Warnings

Log.i(): Information Log.d(): Debugging

Log.v(): Verbose

Η κλάση Log

HelloActivity

```
private static final String TAG = "HelloActivity";

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Intent intent = getIntent();
    String name = intent.getStringExtra("name");

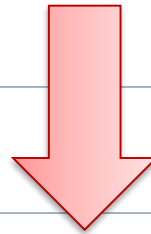
    Log.i(TAG, "The name of the user was retrieved");

    TextView helloTextView = new TextView(this);
    helloTextView.setText("Hello " + name);

    Log.v(TAG, "The hello message was displayed");

    setContentView(helloTextView);
}
```

LogCat Monitor



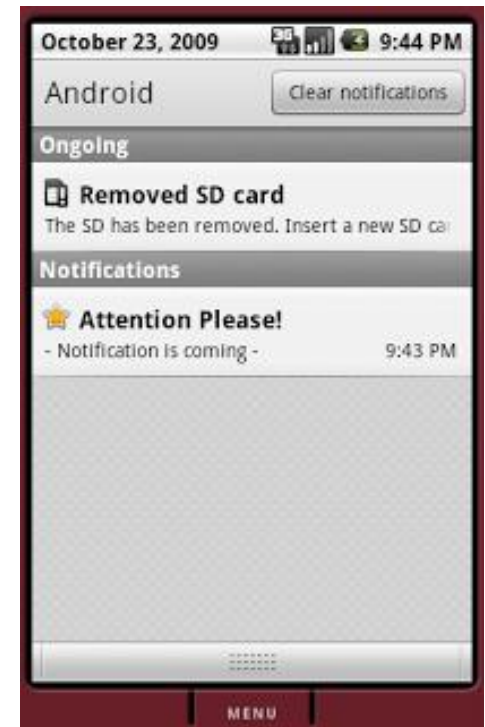
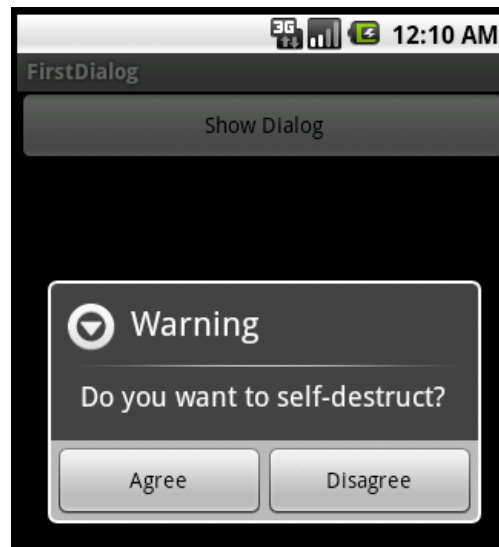
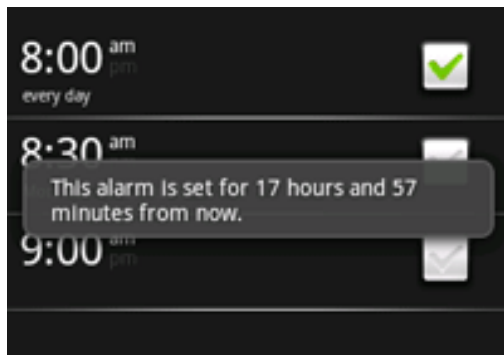
Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.

verbose

L..	Time	PID	TID	Application	Tag	Text
I	12-06 23:02:38.646	670	670	gr.uoa.di.helloandroid	HelloActivity	The name of the user was retrieved
V	12-06 23:02:38.666	670	670	gr.uoa.di.helloandroid	HelloActivity	The hello message was displayed

Ειδοποιήσεις (Notifications) (1)

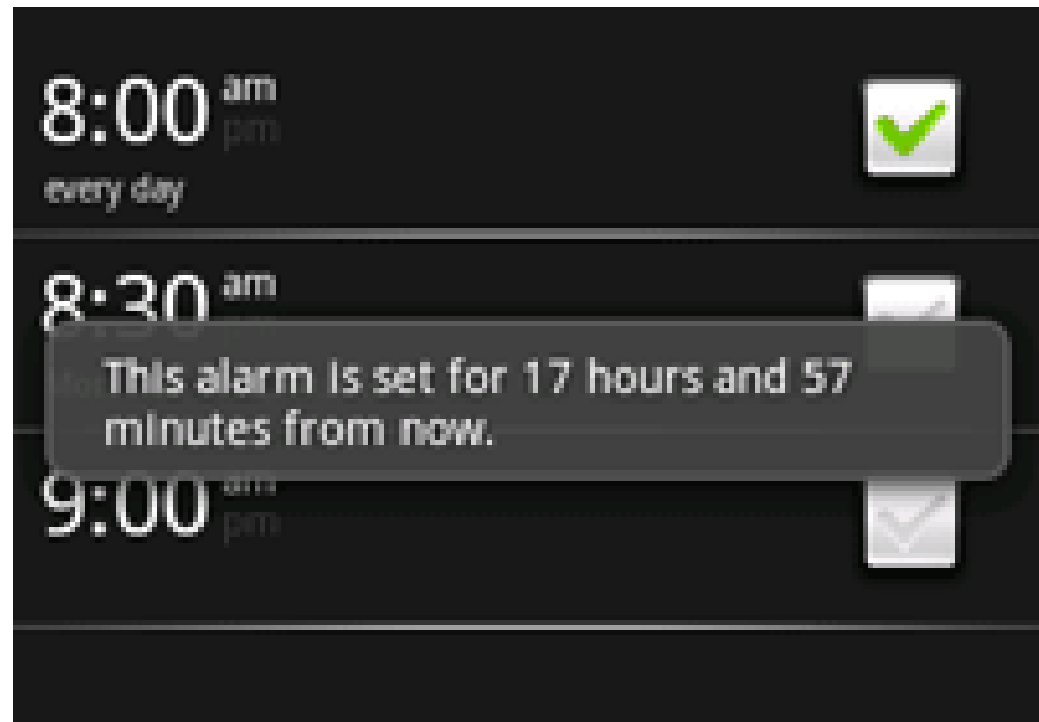
- Δημιουργία ειδοποιήσεων στο Android
- Τριών ειδών ειδοποιήσεις
 - Toast
 - AlertDialog
 - Notification



Ειδοποιήσεις (Notifications) (2)

□ Toast

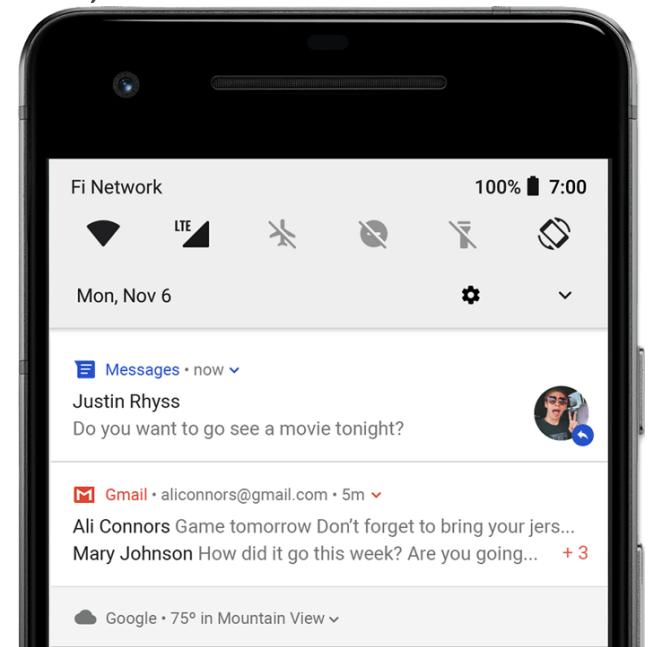
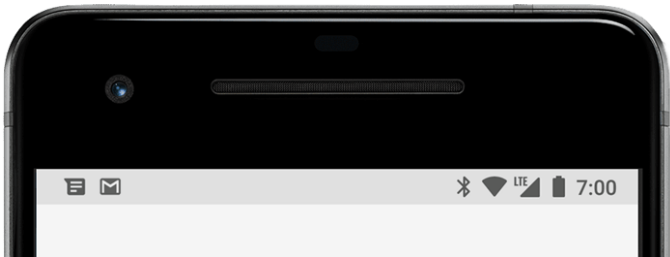
```
Toast toast = Toast.makeText(this,"Notification",Toast.LENGTH_SHORT);  
toast.show();
```



Ειδοποιήσεις (Notifications) (3)

□ Notification

```
NotificationCompat.Builder mBuilder = new  
NotificationCompat.Builder(this, CHANNEL_ID)  
    .setSmallIcon(R.drawable.notification_icon)  
    .setContentTitle(textTitle)  
    .setContentText(textContent)  
    .setPriority(NotificationCompat.PRIORITY_DEFAULT);
```



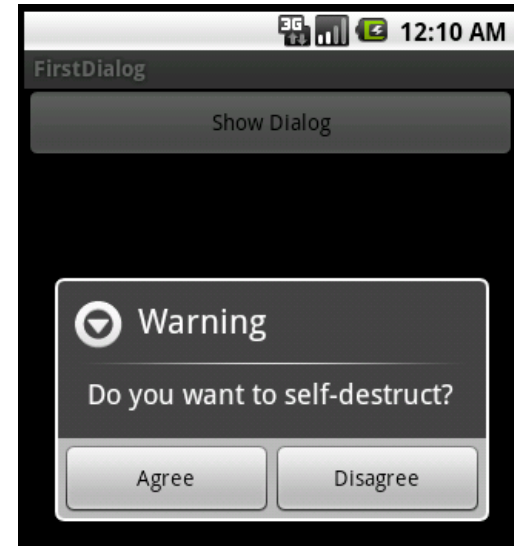
Ειδοποιήσεις (Notifications) (4)

AlertDialog

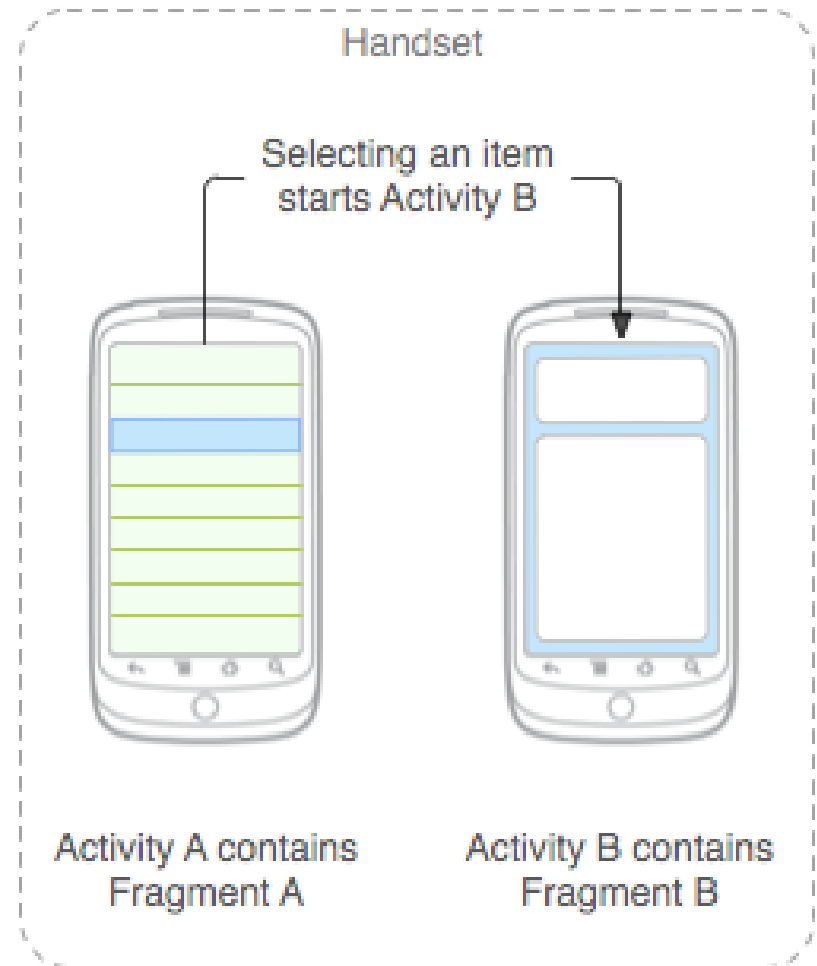
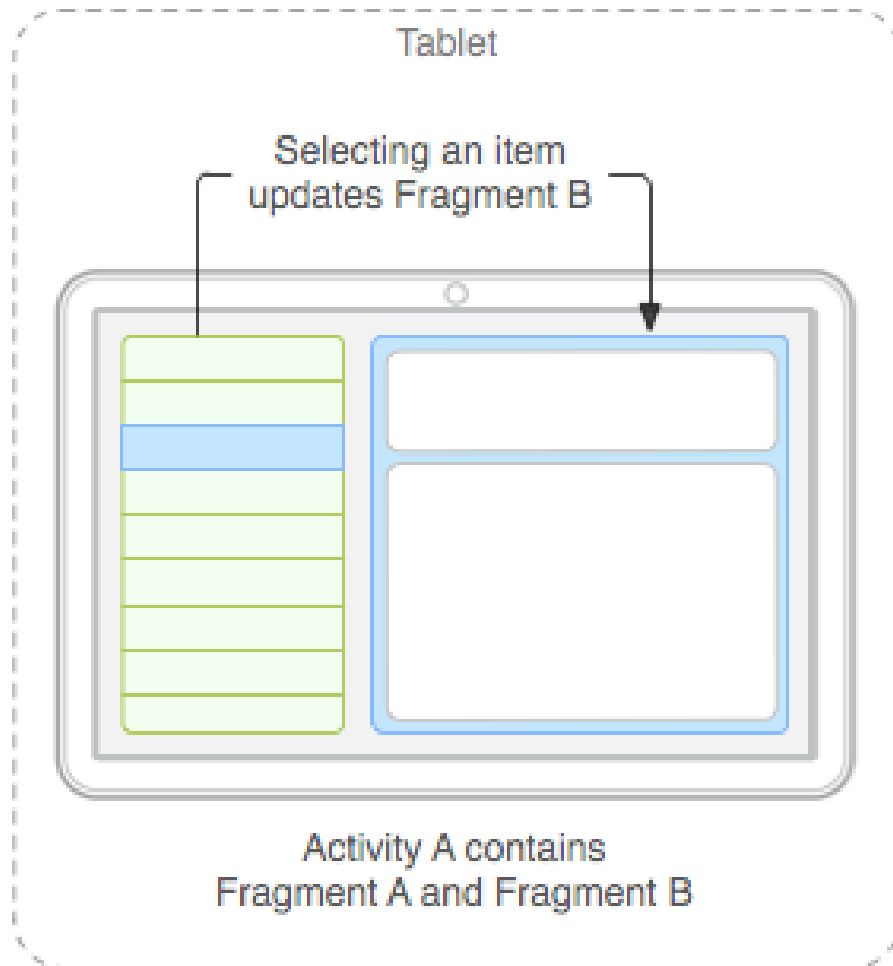
```
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
```

```
builder.setMessage("Do you want to self-destruct?")
    .setPositiveButton("Agree", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            // Exit code
        }
    })
    .setNegativeButton("Disagree", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            // Return code
        }
    });
```

```
AlertDialog dialog = builder.create();
```



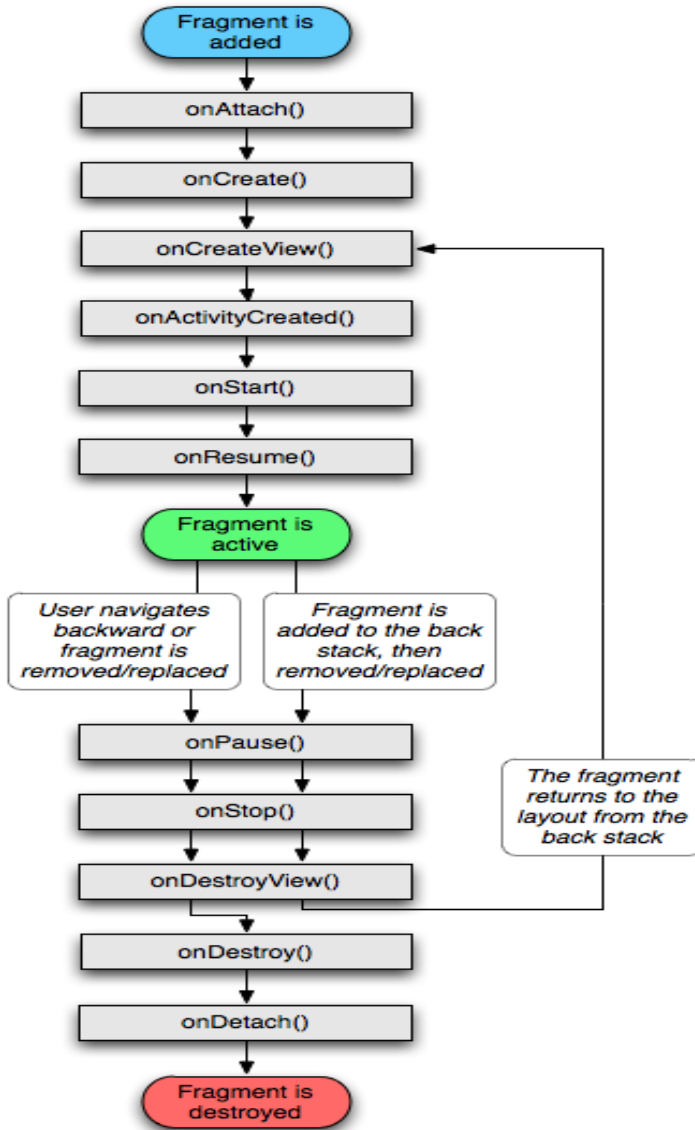
Fragments



Fragments

- ❑ Ανεξάρτητο component το οποίο χρησιμοποιείται από activities.
 - Θεωρείται ως ένα είδος sub-activity
- ❑ Για τα fragments ορίζεται κύκλος ζωής αντίστοιχος του κύκλου ζωής ενός activity
- ❑ Δυναμικός και στατικός ορισμός (on application running)
- ❑ Μια κλάση ορίζεται ως Fragment
 - **extend Fragment, ListFragment, DialogFragment, PreferenceFragment**
 - **Override onCreateView()**
- ❑ Γενικά μας δίνει την δυνατότητα να επαναχρησιμοποιούμε components σε διαφορετικά layouts

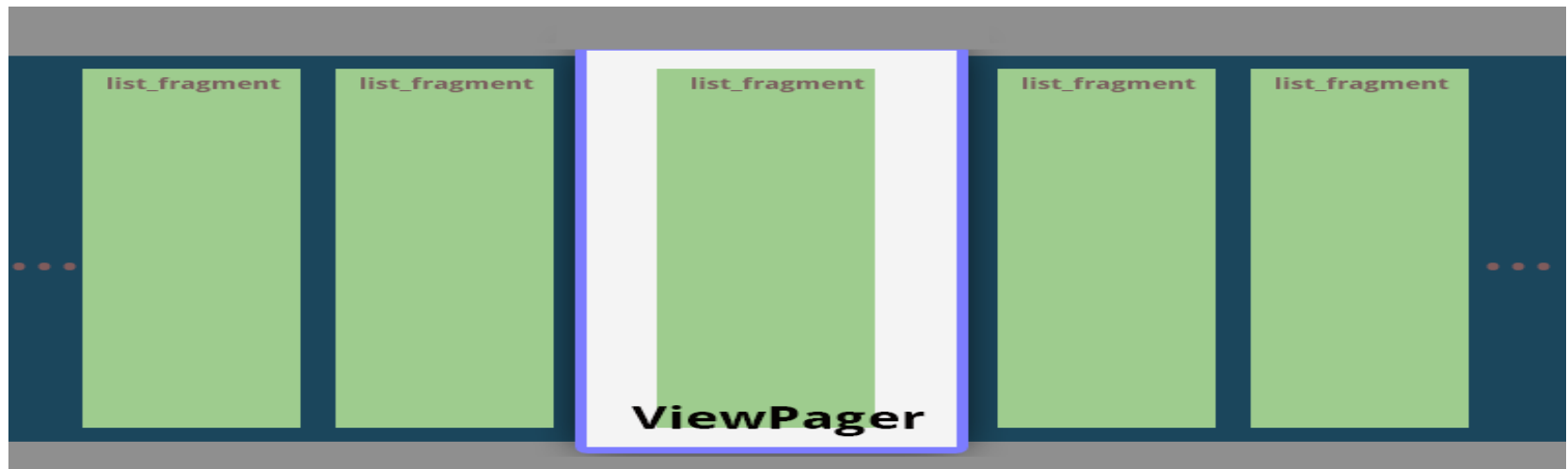
Fragments



Method	Description
onAttach()	To instance του fragment συσχετίζεται με το instance του activity.
onCreate()	Δημιουργία του fragment
onCreateView()	To instance του fragment δημιουργεί την ιεραρχία παρουσίασης (view hierarchy) του. Η ιεραρχία αυτή γίνεται μέλος της ιεραρχίας του activity στο οποίο έχει οριστεί το fragment.
onActivityCreated()	To activity και το fragment έχουν δημιουργήσει τα instances τους με βάση τα view hierarchy τους. Πλέον υπάρχει πρόσβαση σε αυτά με την μέθοδο findViewById() .
onResume()	To fragment είναι πλέον ενεργό και ορατό στον χρήστη.
onPause()	To fragment είναι ορατό αλλά όχι ενεργό, πχ κάποιο άλλο activity χρησιμοποιεί το συγκεκριμένο fragment.
onStop()	To fragment πλέον δεν είναι ορατό στον χρήστη.

Κλάση ViewPager

- ❑ Layout διαχειριστής, επιτρέπει στον χρήστη να πλοηγείται «δεξιά - αριστερά» σε διαφορετικές οθόνες δεδομένων.
- ❑ Η χρήση του ViewPager προϋποθέτει τον ορισμό του PageAdapter. Ο PageAdapter είναι υπεύθυνος για την δημιουργία των οθονών του ViewPager.
- ❑ Συνδυαστική χρήση με Fragment με σκοπό την βέλτιστη διαχείριση του κύκλου ζωής της εκάστοτε σελίδας.



ΠΕΡΑΙΤΕΡΩ ΜΕΛΕΤΗ

Android API Guides

The screenshot shows the Android API Guides website. The top navigation bar is green with the Android logo and the word 'Developers'. Below it, there are tabs for 'DESIGN', 'DEVELOP', and 'DISTRIBUTE'. A search bar is located in the top right corner. The left sidebar is dark grey and contains a list of API Guide categories, with 'Introduction' selected. The main content area is white and features the title 'Introduction to Android' and a breadcrumb 'Develop > API Guides'. The text explains that Android provides a rich application framework for building apps and games in a Java language environment. It also mentions that for new developers, understanding fundamental concepts is important. Two callout boxes on the right provide additional resources: 'To learn how apps work, start with App Fundamentals.' and 'To begin coding right away, read Building Your First App.' Below the main text, there are two sections: 'Apps provide multiple entry points' and 'Apps adapt to different devices', each with a 'Learn more:' link.

Developers

DESIGN DEVELOP DISTRIBUTE

Search

← API Guides

Introduction

App Fundamentals

Device Compatibility

System Permissions

Platform Architecture

App Components

App Resources

App Manifest

User Interface

Animation and Graphics

Computation

Media Apps

Media and Camera

Location and Sensors

Connectivity

Text and Input

Data Storage

Administration

Web Apps

Best Practices

Develop > API Guides

Introduction to Android

Android provides a rich application framework that allows you to build innovative apps and games for mobile devices in a Java language environment. The documents listed in the left navigation provide details about how to build apps using Android's various APIs.

If you're new to Android development, it's important that you understand the following fundamental concepts about the Android app framework:

Apps provide multiple entry points

Android apps are built as a combination of distinct components that can be invoked individually. For instance, an individual *activity* provides a single screen for a user interface, and a *service* independently performs work in the background.

From one component you can start another component using an *intent*. You can even start a component in a different app, such as an activity in a maps app to show an address. This model provides multiple entry points for a single app and allows any app to behave as a user's "default" for an action that other apps may invoke.

Learn more:

- [App Fundamentals](#)
- [Intents and Intent Filters](#)
- [Activities](#)

Apps adapt to different devices

Android provides an adaptive app framework that allows you to provide unique resources for different device configurations. For example, you can create different XML layout files for different screen sizes and the system determines which layout to apply based on the current device's screen size.

You can query the availability of device features at runtime if any app features require specific hardware such as a camera. If necessary, you can also declare features your app requires so app markets such as Google Play Store do not allow installation on devices that do not support that feature.

Learn more:

- [Device Compatibility](#)
- [Resources Overview](#)
- [User Interface Overview](#)

To learn how apps work, start with [App Fundamentals](#).

To begin coding right away, read [Building Your First App](#).

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

Οδηγίες για την πρώτη σας εφαρμογή

The screenshot shows the Android Developers website interface. At the top, there is a green navigation bar with the Android logo and the word 'Developers'. Below this, there are three main sections: 'DESIGN', 'DEVELOP', and 'DISTRIBUTE'. A search bar is located on the right side of the top bar. On the left side, there is a dark grey sidebar with a 'Training' section. The main content area is white and features a breadcrumb trail: 'Develop > Training > Getting Started'. The title of the page is 'Building Your First App'. Below the title, there is a welcome message: 'Welcome to Android application development!'. This is followed by a paragraph explaining the class: 'This class teaches you how to build your first Android app. You'll learn how to create an Android project with Android Studio and run a debuggable version of the app. You'll also learn some fundamentals of Android app design, including how to build a simple user interface and handle user input.' Below this paragraph, there is a note: 'Before you start this class, download and install [Android Studio](#).' On the right side of the main content area, there are two boxes. The top one contains the text 'Get started >'. The bottom one contains the text 'Dependencies' followed by a link '> [Android Studio](#)'. At the bottom of the main content area, there is a large white box with the text 'Get started →' in blue.

Developers

DESIGN DEVELOP DISTRIBUTE

Search

← Training

Getting Started ^

Building Your First App ^

Create an Android Project

Run Your App

Build a Simple User Interface

Start Another Activity

Supporting Different Devices v

Building a Dynamic UI with Fragments v

Saving Data v

Interacting with Other Apps v

Working with System Permissions v

Develop > Training > Getting Started

Building Your First App

Welcome to Android application development!

This class teaches you how to build your first Android app. You'll learn how to create an Android project with Android Studio and run a debuggable version of the app. You'll also learn some fundamentals of Android app design, including how to build a simple user interface and handle user input.

Before you start this class, download and install [Android Studio](#).

Get started >

Dependencies

> [Android Studio](#)

Get started →

Σύνδεσμος: <https://developer.android.com/training/basics/firstapp/index.html>

-
- Εισαγωγή σε Android
 - Δομή αρχείων σε Android Projects
 - Activity
 - Intent
 - Άλλες σημαντικές κλάσεις
 - **Sensors**

Αισθητήρες - Sensors

- Τι είναι αισθητήρας;
 - Συσκευή που ανιχνεύει και αποκρίνεται σε κάποιου είδους εξωτερικό ερέθισμα
- Τύποι αισθητήρων
 - Κίνησης (π.χ. επιταχυνσιόμετρο, γυροσκόπιο)
 - Περιβάλλοντος (π.χ. θερμοκρασίας, φωτισμού)
 - Τοποθεσίας (π.χ. προσανατολισμού, μαγνητόμετρο)

Sensor	Type	Description	Common Uses
<u>TYPE_ACCELEROMETER</u>	Hardware	Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z), including the force of gravity.	Motion detection (shake, tilt, etc.).
<u>TYPE_AMBIENT_TEMPERATURE</u>	Hardware	Measures the ambient room temperature in degrees Celsius ($^{\circ}C$). See note below.	Monitoring air temperatures.
<u>TYPE_GRAVITY</u>	Software or Hardware	Measures the force of gravity in m/s^2 that is applied to a device on all three physical axes (x, y, z).	Motion detection (shake, tilt, etc.).
<u>TYPE_GYROSCOPE</u>	Hardware	Measures a device's rate of rotation in rad/s around each of the three physical axes (x, y, and z).	Rotation detection (spin, turn, etc.).
<u>TYPE_LIGHT</u>	Hardware	Measures the ambient light level (illumination) in lx.	Controlling screen brightness.
<u>TYPE_LINEAR_ACCELERATION</u>	Software or Hardware	Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z), excluding the force of gravity.	Monitoring acceleration along a single axis.
<u>TYPE_MAGNETIC_FIELD</u>	Hardware	Measures the ambient geomagnetic field for all three physical axes (x, y, z) in μT .	Creating a compass.
<u>TYPE_ORIENTATION</u>	Software	Measures degrees of rotation that a device makes around all three physical axes (x, y, z). As of API level 3 you can obtain the inclination matrix and rotation matrix for a device by using the gravity sensor and the geomagnetic field sensor in conjunction with the <u>getRotationMatrix()</u> method.	Determining device position.

Sensor	Type	Description	Common Uses
<u>TYPE_PRESSURE</u>	Hardware	Measures the ambient air pressure in hPa or mbar.	Monitoring air pressure changes.
<u>TYPE_PROXIMITY</u>	Hardware	Measures the proximity of an object in cm relative to the view screen of a device. This sensor is typically used to determine whether a handset is being held up to a person's ear.	Phone position during a call.
<u>TYPE_RELATIVE_HUMIDITY</u>	Hardware	Measures the relative ambient humidity in percent (%).	Monitoring dewpoint, absolute, and relative humidity.
<u>TYPE_ROTATION_VECTOR</u>	Software or Hardware	Measures the orientation of a device by providing the three elements of the device's rotation vector.	Motion detection and rotation detection.
<u>TYPE_TEMPERATURE</u>	Hardware	Measures the temperature of the device in degrees Celsius (°C). This sensor implementation varies across devices and this sensor was replaced with the <u>TYPE_AMBIENT_TEMPERATURE</u> sensor in API Level 14	Monitoring temperatures.

Android sensor framework (ASF)

- Μπορεί κανείς να έχει πρόσβαση στους διαθέσιμους αισθητήρες και να αποκτήσει raw δεδομένα χρησιμοποιώντας το ASF
- Το ASF παρέχει κλάσεις και διεπαφές που βοηθούν στην εκτέλεση εργασιών σχετικών με τους αισθητήρες
 - Προσδιορίζουν ποιοι αισθητήρες είναι διαθέσιμοι σε μια συσκευή.
 - Προσδιορίζουν τις δυνατότητες μεμονωμένων αισθητήρων, όπως μέγιστη εμβέλεια, κατασκευαστή, απαιτήσεις ισχύος και ανάλυση.
 - Αποκτούν δεδομένα αισθητήρων και καθορίζουν τον ελάχιστο ρυθμό απόκτησης των δεδομένων αισθητήρων.
 - Καταχωρούν και καταργούν sensor event listeners που παρακολουθούν τις αλλαγές αισθητήρα

Android Sensors API

□ Sensor Manager

- System service; gives access to hardware sensors

□ Sensor

- Representation of a sensor in a device

□ SensorEventListener

- Interface providing callbacks

□ SensorEvent

- Data structure with event information
-

SensorManager

- Μπορείτε να χρησιμοποιήσετε αυτήν την κλάση για να δημιουργήσετε ένα instance ενός **sensor service**. Αυτή η κλάση παρέχει διάφορες μεθόδους για
 - την πρόσβαση και δημιουργία της λίστας των διαθέσιμων αισθητήρων,
 - την καταχώριση και την κατάργηση sensor event listeners και
 - την απόκτηση πληροφοριών προσανατολισμού.
- Αυτή η κλάση παρέχει επίσης αρκετές σταθερές αισθητήρων οι οποίες χρησιμοποιούνται για να αναφέρουν την ακρίβεια του αισθητήρα, για τον ορισμό των ρυθμών λήψης δεδομένων και για την βαθμονόμηση των αισθητήρων.

Sensor

- Μπορείτε να χρησιμοποιήσετε αυτήν την κλάση για να δημιουργήσετε το **instance ενός συγκεκριμένου αισθητήρα**.
 - Αυτή η κλάση παρέχει διάφορες μεθόδους που σας επιτρέπουν να προσδιορίσετε τις δυνατότητες ενός αισθητήρα.

SensorEvent

- Το σύστημα χρησιμοποιεί αυτή την κλάση για να δημιουργήσει ένα **αντικείμενο sensor event**, το οποίο παρέχει πληροφορίες σχετικά με ένα event αισθητήρα.
 - τα ακατέργαστα δεδομένα αισθητήρα,
 - τον τύπο του αισθητήρα που δημιούργησε το συμβάν,
 - την ακρίβεια των δεδομένων και τη χρονική σφραγίδα για το συμβάν

SensorEventListener

- Μπορείτε να χρησιμοποιήσετε αυτήν διεπαφή για να δημιουργήσετε **δύο callback μεθόδους** που λαμβάνουν ειδοποιήσεις (συμβάντα αισθητήρων) όταν αλλάζουν
 1. οι τιμές του αισθητήρα ή
 2. όταν αλλάζει η ακρίβεια του αισθητήρα.

Πως χρησιμοποιείτε τα API

Σε μια τυπική εφαρμογή χρησιμοποιείτε αυτά τα API που σχετίζονται με αισθητήρες για να εκτελέσετε δύο βασικές εργασίες:

- ❑ **Προσδιορισμός αισθητήρων και δυνατοτήτων τους.** Η αναγνώριση των αισθητήρων και των δυνατοτήτων τους είναι χρήσιμη εάν η εφαρμογή σας διαθέτει χαρακτηριστικά που βασίζονται σε συγκεκριμένους τύπους ή δυνατότητες αισθητήρων. Για παράδειγμα, μπορεί να θέλετε να αναγνωρίσετε όλους τους αισθητήρες που υπάρχουν σε μια συσκευή και να απενεργοποιήσετε τις λειτουργίες εφαρμογής που βασίζονται σε αισθητήρες που δεν υπάρχουν. Ομοίως, μπορεί να θέλετε να εντοπίσετε όλους τους αισθητήρες ενός δεδομένου τύπου, ώστε να μπορείτε να επιλέξετε την εφαρμογή αισθητήρων που έχει τη βέλτιστη απόδοση για την εφαρμογή σας.
- ❑ **Παρακολούθηση συμβάντων αισθητήρων.** Είναι ο τρόπος με τον οποίο αποκτάτε ακατέργαστα δεδομένα αισθητήρα. **Ένα συμβάν αισθητήρα συμβαίνει κάθε φορά που ένας αισθητήρας ανιχνεύει μια αλλαγή στις παραμέτρους που μετράει.** Ένα συμβάν αισθητήρα σας παρέχει τέσσερα τεμάχια πληροφοριών: το όνομα του αισθητήρα που ενεργοποίησε το συμβάν, τη χρονική σφραγίδα για το συμβάν, την ακρίβεια του συμβάντος και τα δεδομένα ακατέργαστου αισθητήρα που ενεργοποίησαν το συμβάν.

```
private SensorManager mSensorManager;
private Sensor mSensor;
...
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);

public void onSensorChanged(SensorEvent event){
    // In this example, alpha is calculated as  $t / (t + dT)$ ,
    // where t is the low-pass filter's time-constant and
    // dT is the event delivery rate.

    final float alpha = 0.8;

    // Isolate the force of gravity with the low-pass filter.
    gravity[0] = alpha * gravity[0] + (1 - alpha) * event.values[0];
    gravity[1] = alpha * gravity[1] + (1 - alpha) * event.values[1];
    gravity[2] = alpha * gravity[2] + (1 - alpha) * event.values[2];

    // Remove the gravity contribution with the high-pass filter.
    linear_acceleration[0] = event.values[0] - gravity[0];
    linear_acceleration[1] = event.values[1] - gravity[1];
    linear_acceleration[2] = event.values[2] - gravity[2];
}
```

Sensor Rates

- ❑ `SensorManager.SENSOR_DELAY_NORMAL` (delay 200000 microseconds) (default value)
- ❑ `SensorManager.SENSOR_DELAY_GAME` (delay 20000 microseconds)
- ❑ `SensorManager.SENSOR_DELAY_UI` (delay 60000 microseconds)
- ❑ `SensorManager.SENSOR_DELAY_FASTEST` (delay 0 microseconds)

Καλές πρακτικές

- ❑ Unregister sensor listeners
- ❑ Don't block the `onSensorChanged()` method
- ❑ Avoid using deprecated methods or sensor types
- ❑ Verify sensors before you use them
- ❑ Choose sensor delays carefully

Ερωτήσεις ;

TabHost

- Αποτελείται από δύο «παιδιά» : Ένα σύνολο από tab labels τα οποία ο χρήστης επιλέγει ώστε να διαλέξει ένα συγκεκριμένο tab καθώς και ένα FrameLayout με πληροφορίες για το περιεχόμενο του κάθε tab.
- Κάποιες βασικές μέθοδοι της κλάσης TabHost είναι :
 - A) `addTab()`
 - B) `getCurrentTab()`
 - C) `getTabContentView()` : Επιστρέφει το FrameLayout το οποίο έχει αποθηκευμένο το περιεχόμενο του tab.
 - D) `setup()` : Καλείται πριν την `addTab()` σε περίπτωση που το tab έχει γίνει load με την χρήση της `findViewById()`.
- ```
mTabHost = (TabHost) findViewById(R.id.tabhost);
mTabHost.setup();
mTabHost.addTab(TAB_TAG_1, "Hello, world!", "Tab 1");
```

# Αναφορές

---

- Android Official Training:

<http://developer.android.com/training/>

- Android Documentation:

<http://developer.android.com/reference/packages.html>

- Android Permissions

<http://developer.android.com/guide/topics/security/permissions.html>

- Android Sensor Overview

[https://developer.android.com/guide/topics/sensors/sensors\\_overview.html](https://developer.android.com/guide/topics/sensors/sensors_overview.html)

# Αναφορές

---

- ❑ Log documentation:

<http://developer.android.com/reference/android/util/Log.html>

- ❑ Android Web Services: <http://mobileorchard.com/android-appdevelopment-calling-web-services/>

- ❑ Android Tab Layout with Swipeable Views :

<http://www.androidhive.info/2013/10/android-tab-layout-withswipeable-views-1/>

- ❑ Fragments:

<http://developer.android.com/guide/components/fragments.html>

- ❑ ViewPager :

<http://developer.android.com/reference/android/support/v4/view/ViewPager.html>

# Βασικά Δομικά Συστατικά Android

---

## □ **Activity**

- Το στρώμα παρουσίασης της εφαρμογής
- Μια εφαρμογή μπορεί να έχει περισσότερα του ενός activities για διαχείριση διαφορετικών φάσεων του προγράμματος
- Κάθε activity είναι υπεύθυνο να αποθηκεύει την δική του κατάσταση

## □ **Intent**

- Μηχανισμός που καθορίζει ποια συγκεκριμένη ενέργεια (action) πρέπει να εκτελεστεί
- Σε android σχεδόν ΟΛΕΣ οι αλληλοεπιδράσεις γίνονται μέσω intents

## □ **Service**

- Διεργασία που τρέχει στο παρασκήνιο χωρίς τη παρέμβαση του χρήστη.
- Δεν παρέχει UI στο χρήστη
- Παρόμοιο με Unix daemon
- Υπάρχουν πολλά built-in services σε Android

## □ **Content Providers**

- Επιτρέπουν σε μια εφαρμογή να έχει πρόσβαση σε δεδομένα άλλων εφαρμογών (π.χ. Contacts)
- Επιτρέπουν σε μια εφαρμογή να διαμοιράζει τα δεδομένα που παράγει σε άλλες εφαρμογές



# Βασικά Δομικά Συστατικά Android

---

## □ Views

- Lists, grids, text boxes, buttons ακόμα και ένας embeddable web browser

## □ Activity Manager

- Διαχειρίζεται τον κύκλο ζωής (life cycle) των εφαρμογών και παρέχει ένα κοινό τρόπο προσπέλασης

## □ Notification Manager

- Επιτρέπει στις εφαρμογές να παρουσιάζουν μηνύματα συναγερμών (alerts) στην γραμμή κατάστασης (status bar)

## □ Resource Manager

- Παρέχει στην εφαρμογή πρόσβαση σε αρχεία εκτός των αρχείων κώδικα, π.χ. localized strings, graphics, layout files

## □ Data storage

- Παρέχει τα δεδομένα στην εφαρμογή και μπορεί να διαμοιραστεί μεταξύ διαφορετικών εφαρμογών. Βάσεις δεδομένων, αρχεία και preferences (hash map) μπορούν να προσπελαστούν από ομάδες εφαρμογών

## □ Broadcast receivers

- Προκαλούν την δημιουργία intents που μπορούν προκαλέσουν την εκτέλεση κώδικα