

Android Development

Een introductie



Intro

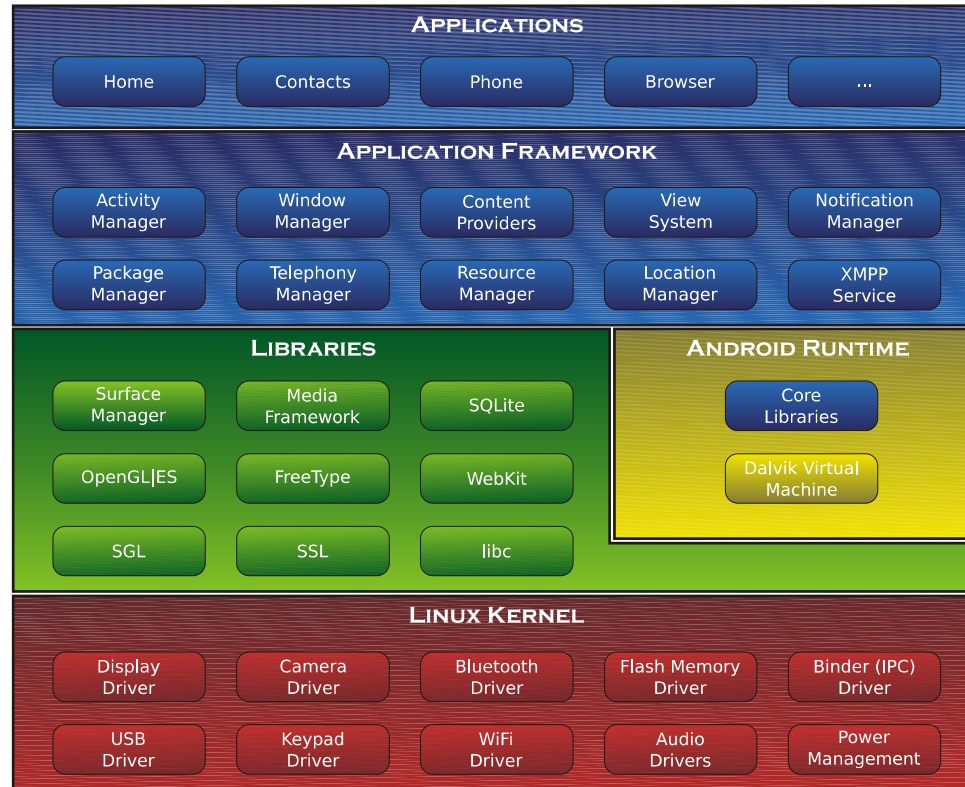
- Xanvier Brouwer
- Consultant, Software Engineer
- 2016 – 2017 afgestudeerd HHS
 - Minor KB-50 (Android Development)
- Young Professional Programma Quintor



Programma

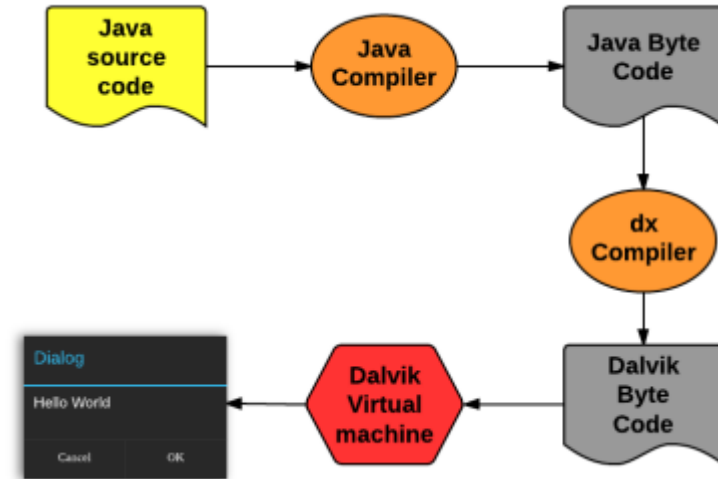
- Algemene Android informatie
- *Hands-on – Opstarten Applicatie*
- User interface, Intents, Android structuur
- *Hands-on – Implementatie UI*
- **Pauze (15 min)**
- Listviews en RecyclerViews
- *Hands-on – RecyclerView*
- Threads & AsyncTask
- *Hands-on – Concurrency en REST*

Android OS



Dalvik VM

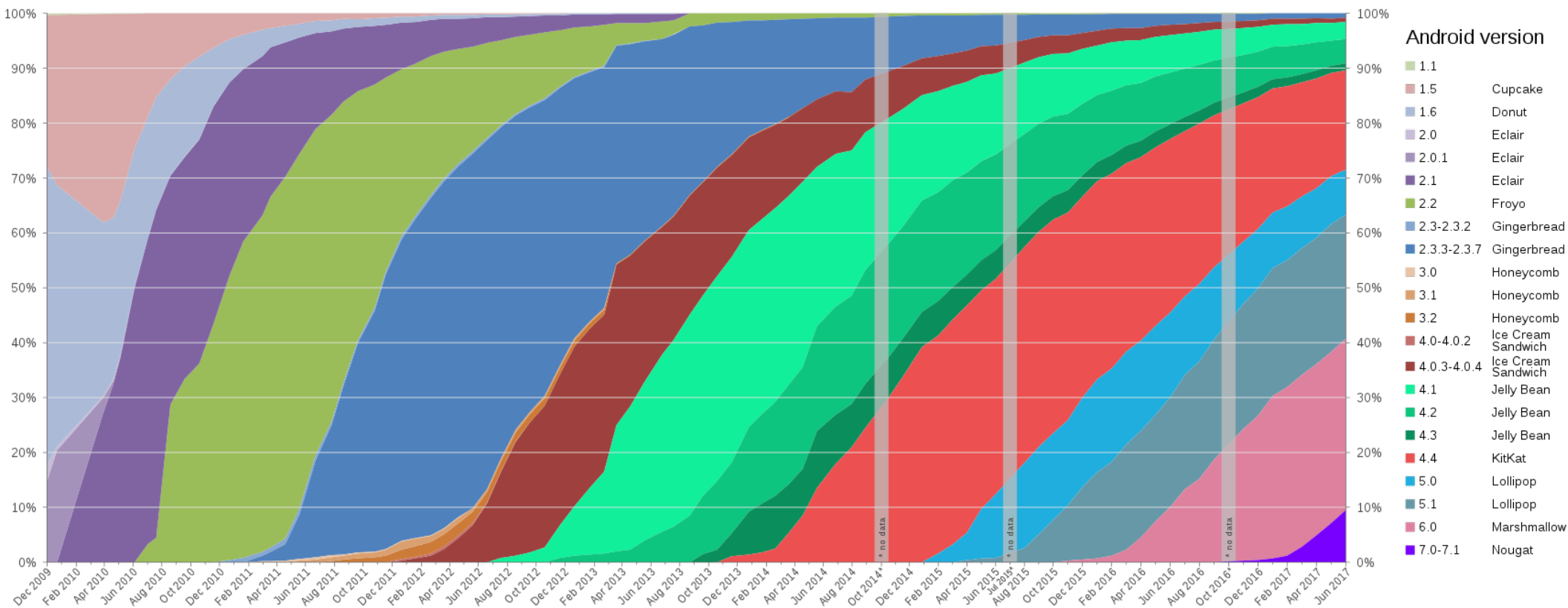
- Highly CPU-optimized
- Custom designed



SDK versions

Code name	Version	Release date	API level	Security
(No codename)	1.0	September 23, 2008	1	Unsupported
"Petit Four"	1.1	February 9, 2009	2	Unsupported
Cupcake	1.5	April 27, 2009	3	Unsupported
Donut	1.6	September 15, 2009	4	Unsupported
Eclair	2.0 – 2.1	October 26, 2009	5 – 7	Unsupported
Froyo	2.2 – 2.2.3	May 20, 2010	8	Unsupported
Gingerbread	2.3 – 2.3.7	December 6, 2010	9 – 10	Unsupported
Honeycomb	3.0 – 3.2.6	February 22, 2011	11 – 13	Unsupported
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	14 – 15	Unsupported
Jelly Bean	4.1 – 4.3.1	July 9, 2012	16 – 18	Unsupported
KitKat	4.4 – 4.4.4	October 31, 2013	19 – 20	Unsupported
Lollipop	5.0 – 5.1.1	November 12, 2014	21 – 22	Supported
Marshmallow	6.0 – 6.0.1	October 5, 2015	23	Supported
Nougat	7.0 – 7.1.2	August 22, 2016	24 – 25	Supported
Oreo	8.0 – 8.1	August 21, 2017	26 – 27	Supported

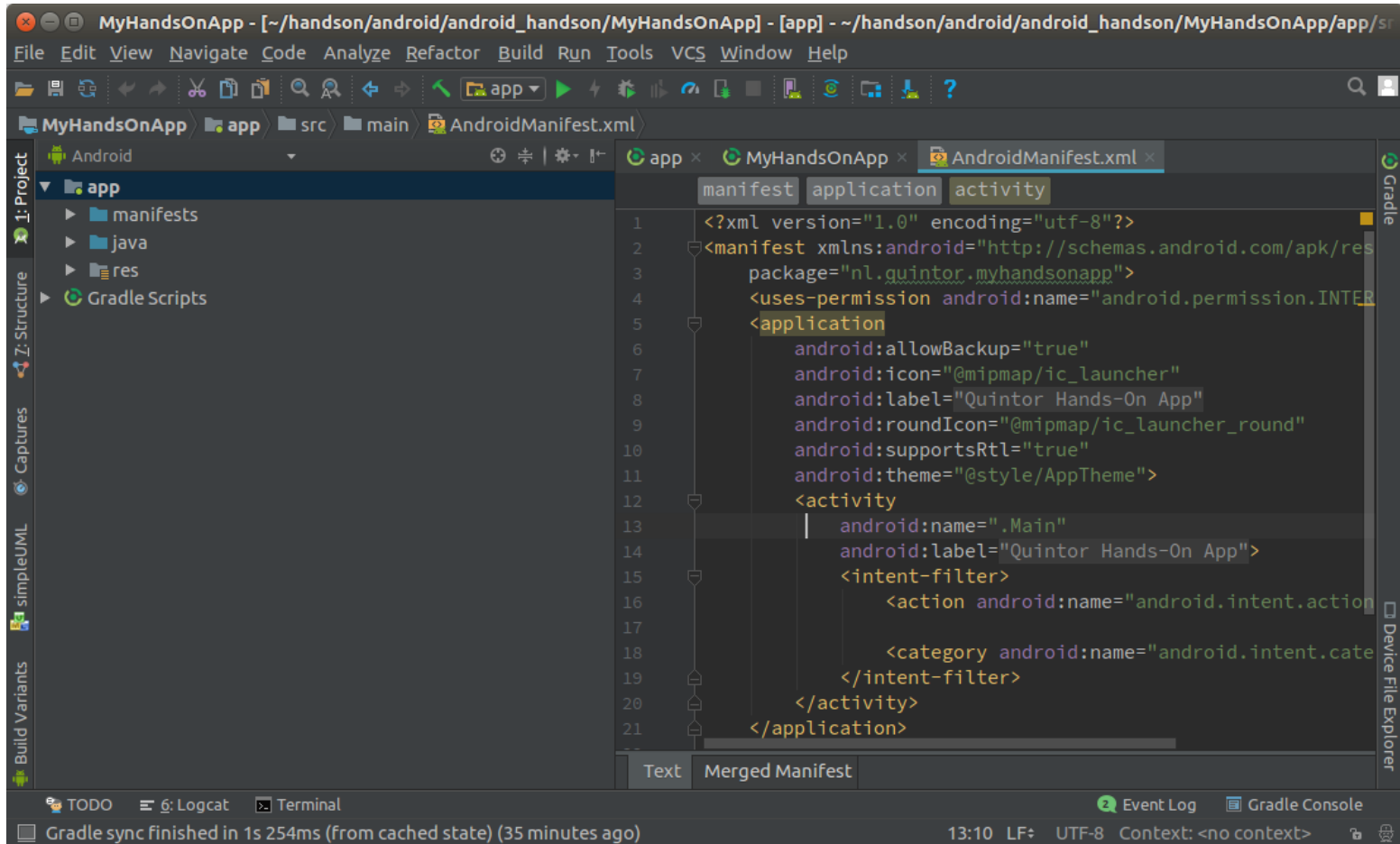
Versie distributie



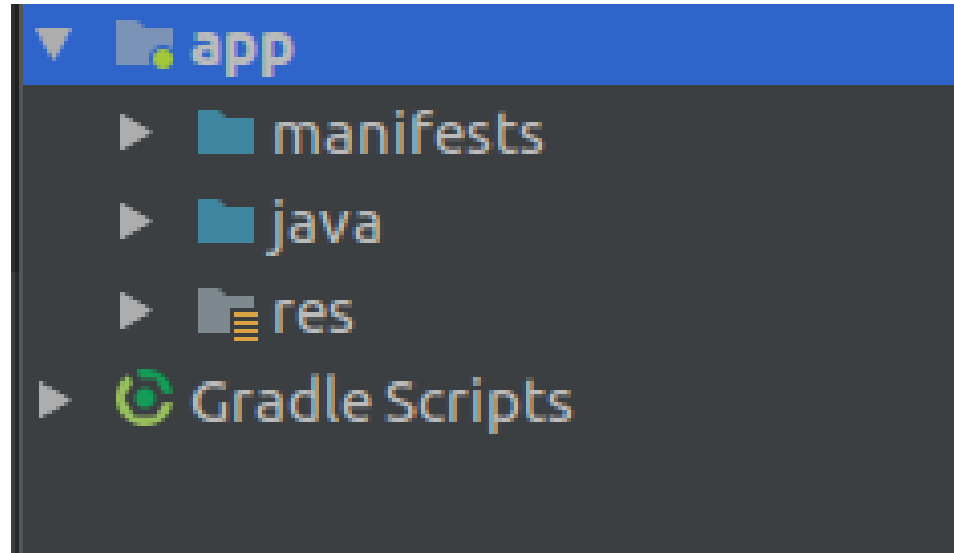
Hardware



Android Development



Applicatie structuur

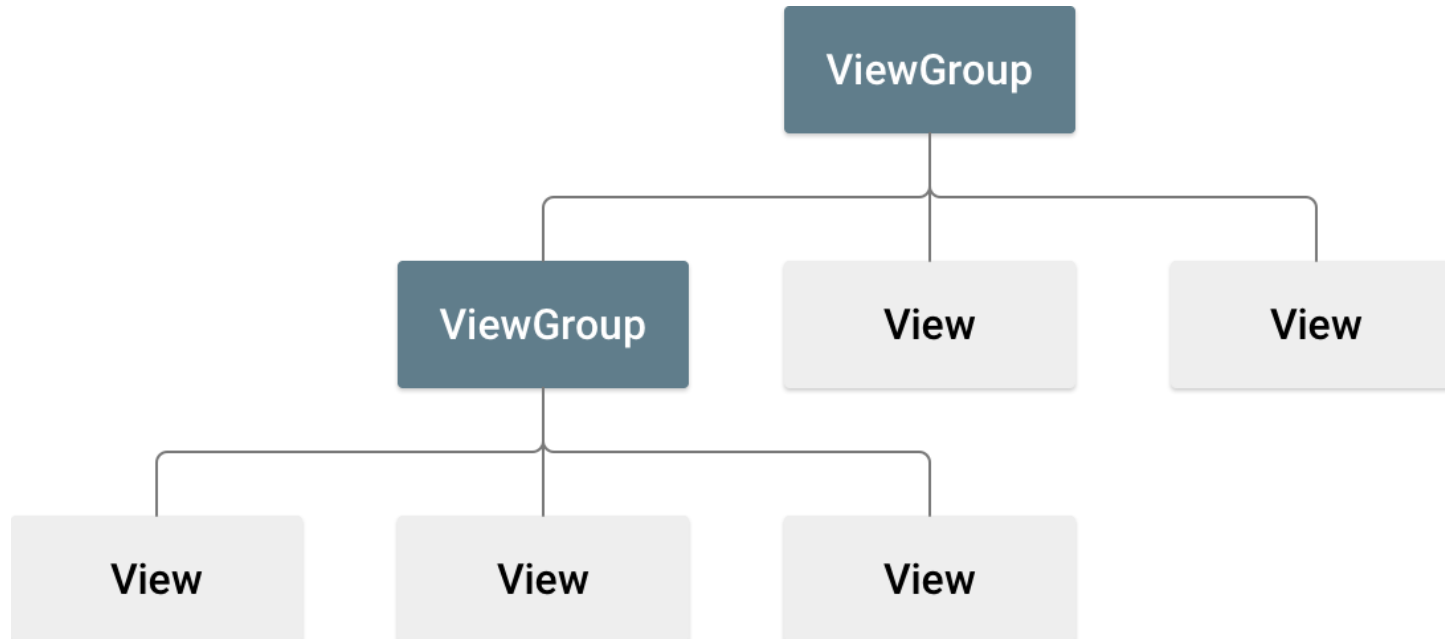


Hands-on

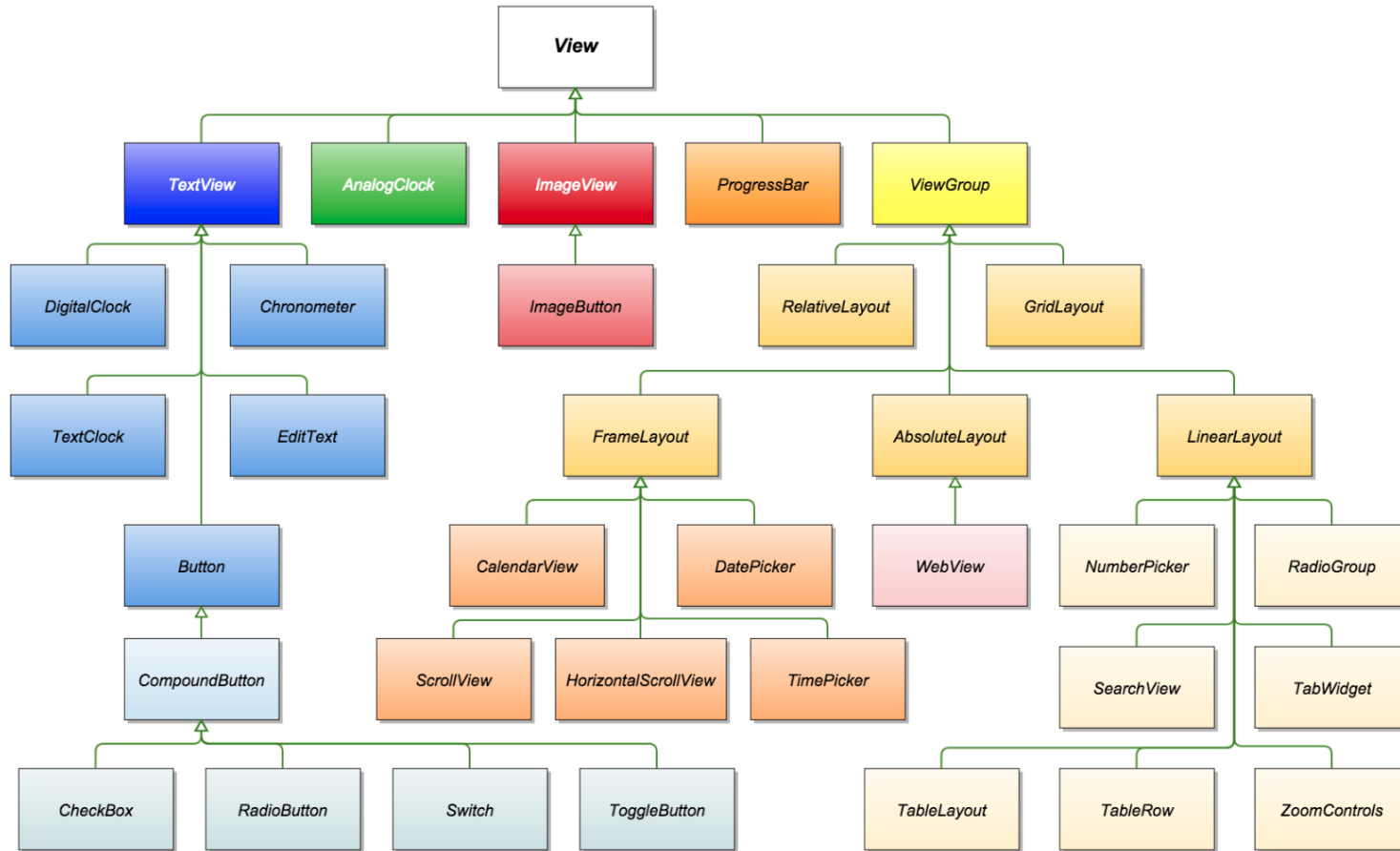
Deel 1

Ga naar https://github.com/xanvierb/android_handson en voer alles uit onder het kopje **Dependencies en eerste start-up**

User Interface



View elements



Activity

- Scherm = Layout XML + Java code

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

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

R-object

- Magic

```
TextView textDisplay =  
    (TextView) findViewById(R.id.text1);
```

View groups

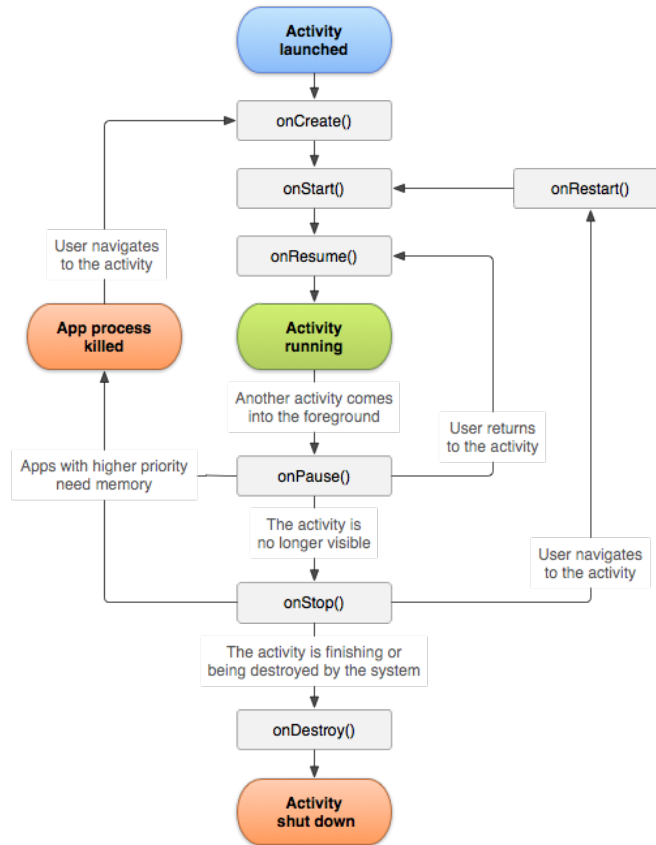


Linear Layout



Relative Layout

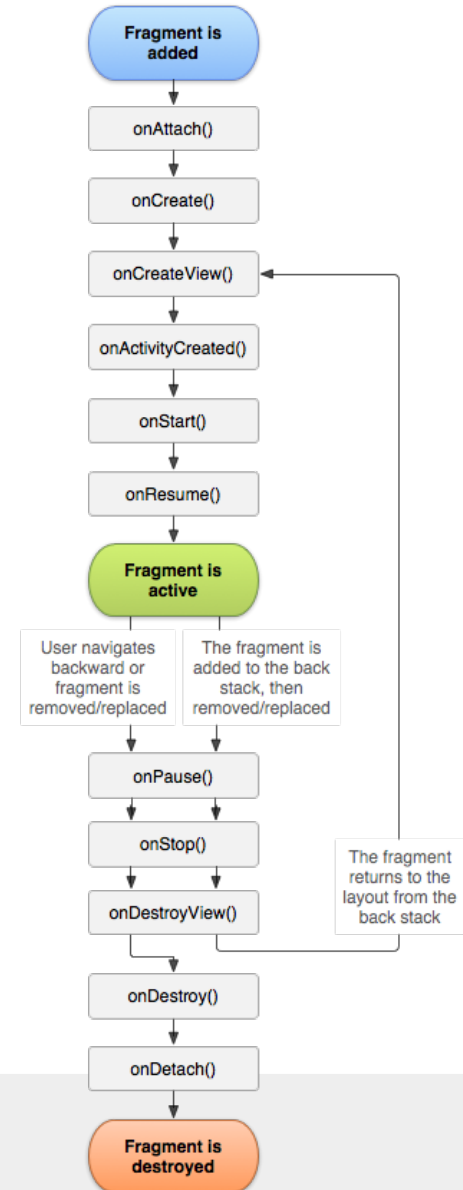
Activity Lifecycle



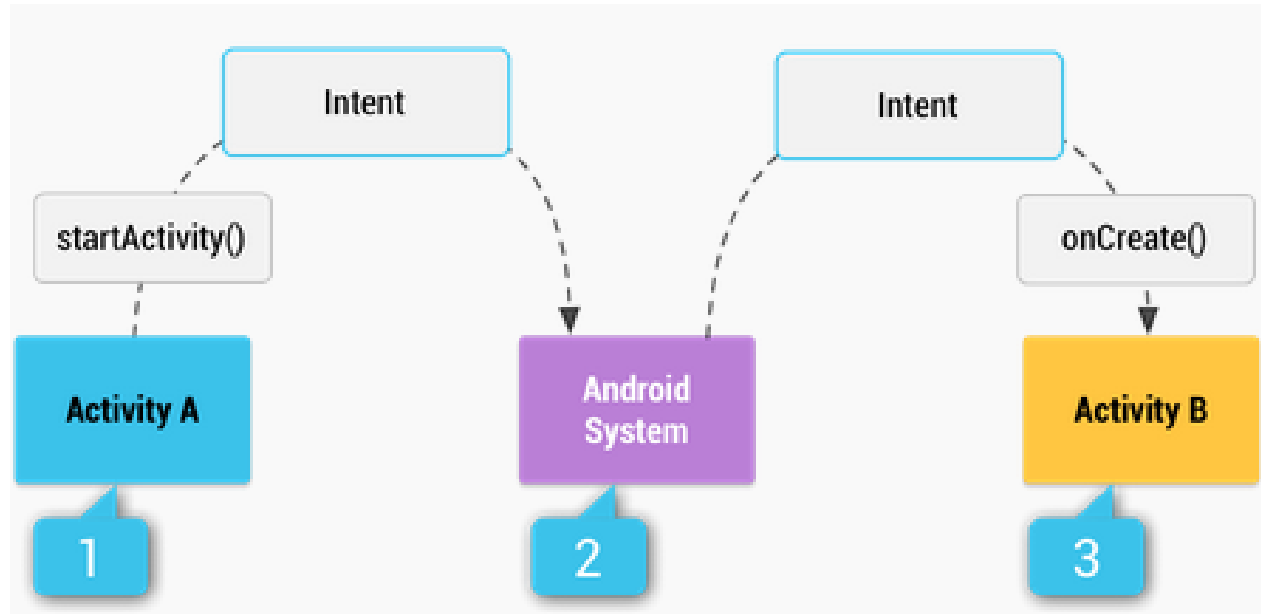
Fragments



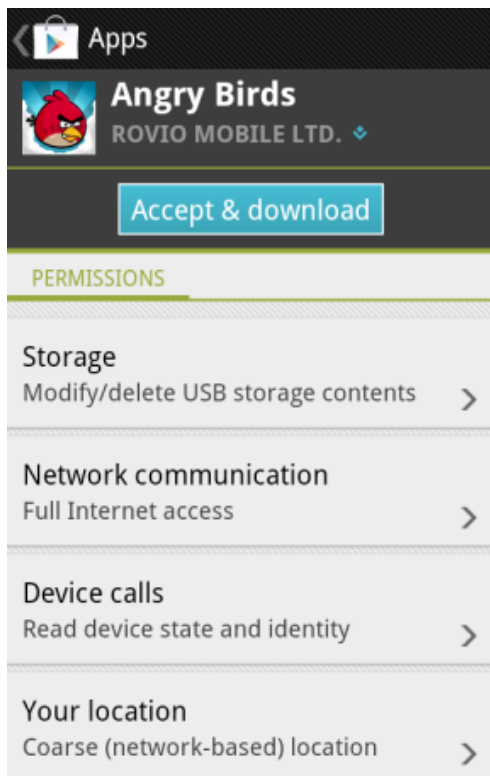
Fragment Lifecycle



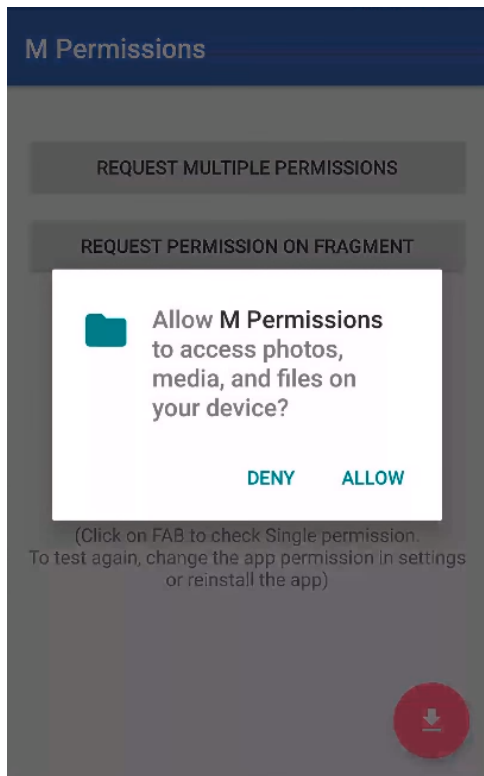
Intents



Permissions



< API 23



>= API 23

Hands-on

Deel 2

Ga naar https://github.com/xanvierb/android_handson en voer alles uit onder het kopje **Android applicatie structuur, User interface en Intents**

Resources

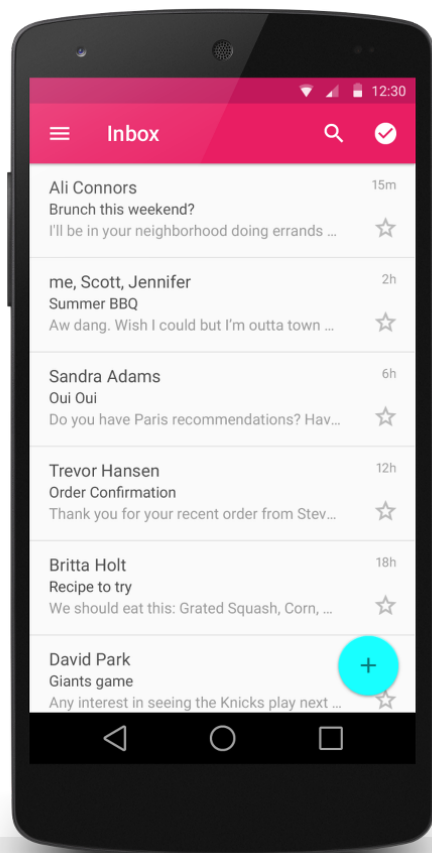
- String
- Drawable/mipmap
- Multi-language?

```
<TextView  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="@string/hello" />
```

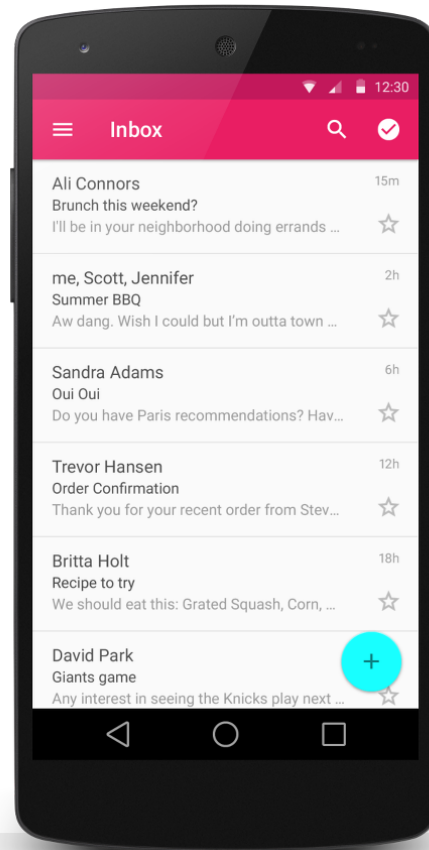
Internationalization (i18n)

- `res/values/strings.xml`
- `res/values-fr/strings.xml`

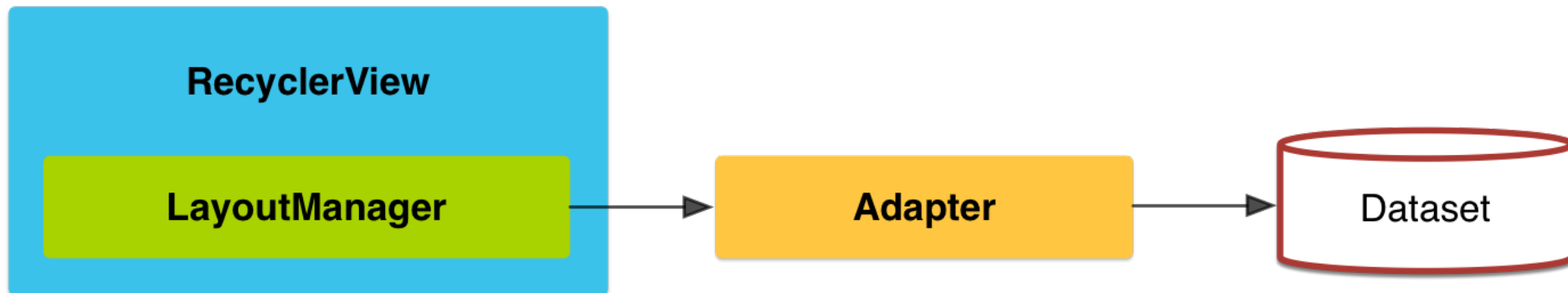
ListViews



~~ListViews~~ RecyclerView



RecyclerView



Hands-on

Deel 3

Ga naar https://github.com/xanvierb/android_handson en voer alles uit onder het kopje **Advanced User Interface**

Data persistentie

- SharedPreferences
- Schrijven naar bestand
- SQLite
 - Room

Threads

- Code uitvoeren op verschillende (virtual) cores
- Applicatie niet blokkeren

Threading op Android

- UI thread
- Blocking
- Regels
 - Blokkeer de UI thread niet
 - Android UI niet benaderen vanaf andere dan de UI Thread
- Oplossing: Worker threads

AsyncTask

```
private class DownloadFilesTask extends AsyncTask<URL, Integer, Long> {  
    protected Long doInBackground(URL... urls) {  
        int count = urls.length;  
        long totalSize = 0;  
        for (int i = 0; i < count; i++) {  
            totalSize += Downloader.downloadFile(urls[i]);  
            publishProgress((int) ((i / (float) count) * 100));  
            // Escape early if cancel() is called  
            if (isCancelled()) break;  
        }  
        return totalSize;  
    }  
  
    protected void onProgressUpdate(Integer... progress) {  
        setProgressPercent(progress[0]);  
    }  
  
    protected void onPostExecute(Long result) {  
        showDialog("Downloaded " + result + " bytes");  
    }  
}
```


REST

- HTTP
- Data ophalen van een API endpoint
- Dit is blocking

Hands-on

Deel 4

Ga naar https://github.com/xanvierb/android_handson en voer alles uit onder het kopje **AsyncTask en REST calls**

Android Development

Een introductie



Xanvier Brouwer

Android Development

Quintor

Intro

- Xanvier Brouwer
- Consultant, Software Engineer
- 2016 – 2017 afgestudeerd HHS
 - Minor KB-50 (Android Development)
- Young Professional Programma Quintor



Xanvier Brouwer

Android Development

Quintor

Programma

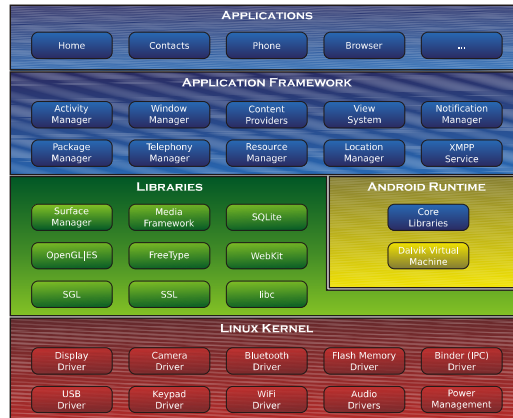
- Algemene Android informatie
- *Hands-on – Opstarten Applicatie*
- User interface, Intents, Android structuur
- *Hands-on – Implementatie UI*
- **Pauze (15 min)**
- Listviews en RecyclerViews
- *Hands-on – RecyclerView*
- Threads & AsyncTask
- *Hands-on – Concurrency en REST*

Xavier Brouwer

Android Development

Quintor

Android OS



Xanvier Brouwer

Android Development

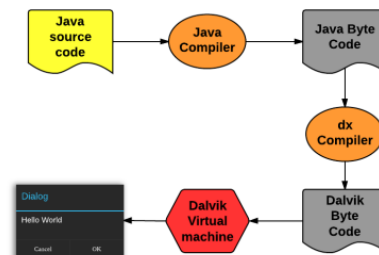
Quintor

Linux kernel provides:

- memory management,
- process isolation,
- threading support,
- Security
- (Pre-processing)
-
- Heeft niet de gebruikelijke GNU libs.
- Geen X-window systeem, daarom geen reguliere applicaties
- Om android applicaties te gebruiken is er de DVM

Dalvik VM

- Highly CPU-optimized
- Custom designed



Xanvier Brouwer

Android Development

Quintor

De Dalvik VM door Dan Bornstein + team

Specifiek en geoptimaliseerd voor embedded gebruik
Grootste voordelen: Java bytecode executeerd 8-bit stack instructies. (lokale variabelen moeten worden gekopieerd) daar de DVM een 16-bit stack heeft en lokale variabelen hier direct kan bewerken.

JAR is groter is formaat tov DEX

SDK versions

Code name	Version	Release date	API level	Security
(No codename)	1.0	September 23, 2008	1	Unsupported
"Petit Four"	1.1	February 9, 2009	2	Unsupported
Cupcake	1.5	April 27, 2009	3	Unsupported
Donut	1.6	September 15, 2009	4	Unsupported
Eclair	2.0 – 2.1	October 26, 2009	5 – 7	Unsupported
Froyo	2.2 – 2.2.3	May 20, 2010	8	Unsupported
Gingerbread	2.3 – 2.3.7	December 6, 2010	9 – 10	Unsupported
Honeycomb	3.0 – 3.2.6	February 22, 2011	11 – 13	Unsupported
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	14 – 15	Unsupported
Jelly Bean	4.1 – 4.3.1	July 9, 2012	16 – 18	Unsupported
KitKat	4.4 – 4.4.4	October 31, 2013	19 – 20	Unsupported
Lollipop	5.0 – 5.1.1	November 12, 2014	21 – 22	Supported
Marshmallow	6.0 – 6.0.1	October 5, 2015	23	Supported
Nougat	7.0 – 7.1.2	August 22, 2016	24 – 25	Supported
Oreo	8.0 – 8.1	August 21, 2017	26 – 27	Supported

Xanvier Brouwer

Android Development

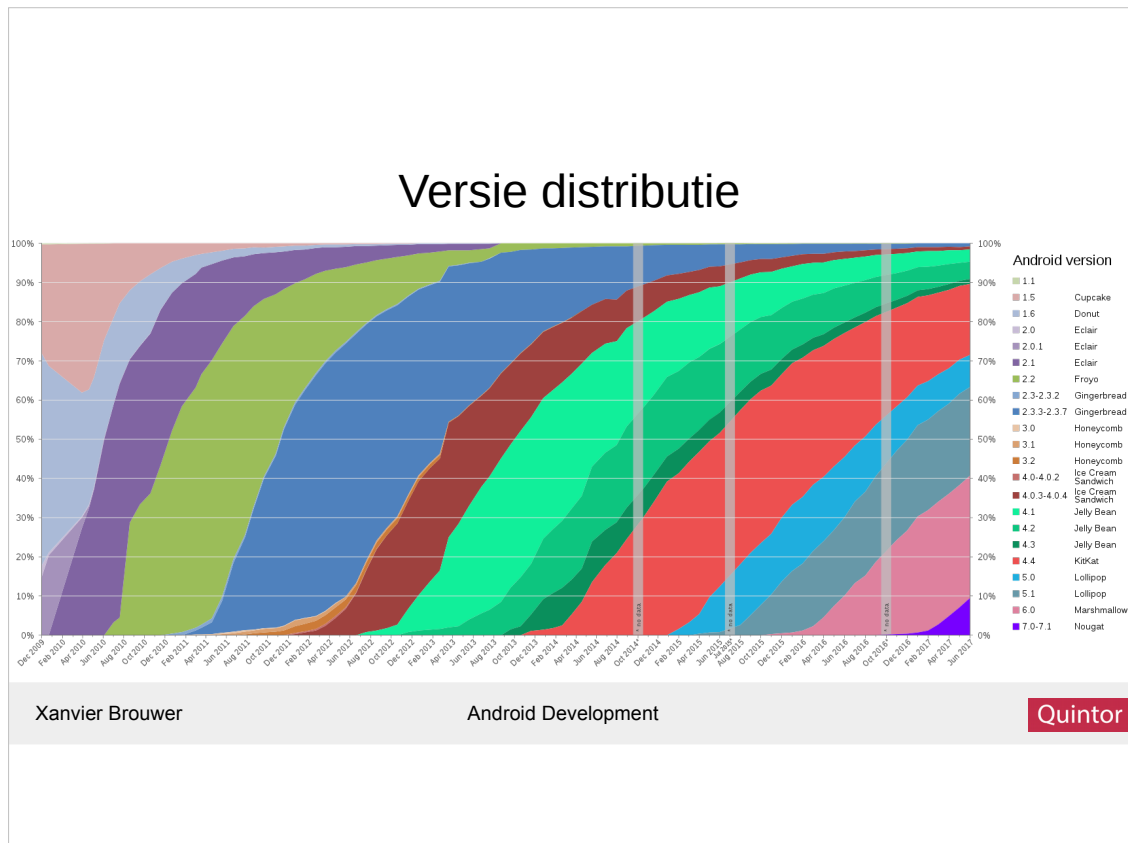
Quintor

Dit zijn alle Android versies.

Laatste 4 worden nu ondersteund met security patches

Android 9 (Android P) wordt waarschijnlijk aangekondigd in maart en ge-released in augustus
Uitdagend om te ontwikkelen voor zoveel verschillende versies

Support libraries



4.4.4, wat niet meer ondersteund wordt is nog steeds in groot gebruik.

Hardware



Xanvier Brouwer

Android Development

Quintor

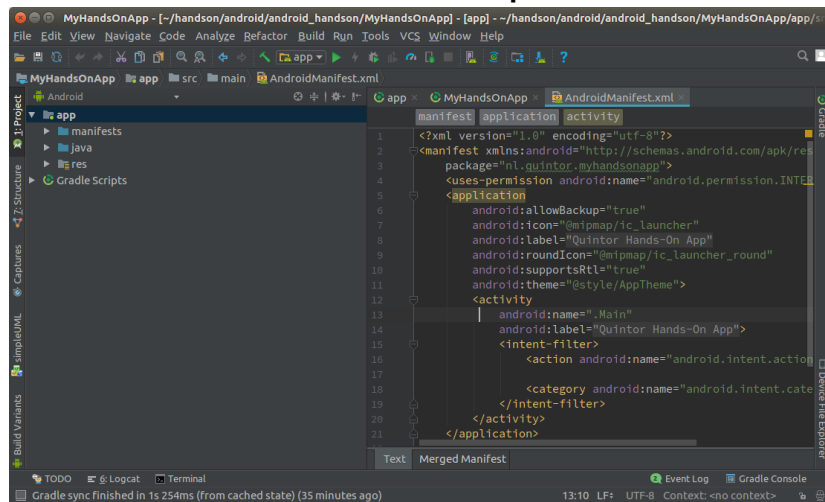
Links:

Alcatel PIXI 4
4 inch scherm
512 mb RAM

Rechts:

Google Pixel 2 XL
6 inch scherm
4 GB geheugen

Android Development



Xanvier Brouwer

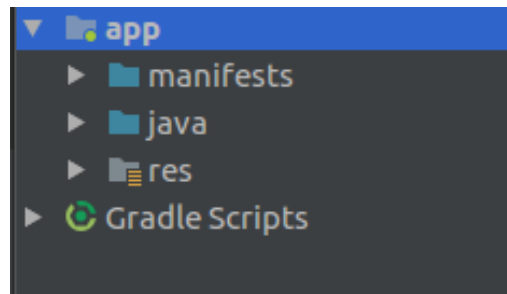
Android Development

Quintor

Vroeger Werd er gebruik gemaakt van Eclipse met plugins voor Android development

Android wordt tegenwoordig ontwikkeld in Android Studio dat is gebaseerd op IntelliJ Community Edition. Een zeer complete tool met alles er op en eraan.

Applicatie structuur



Xanvier Brouwer

Android Development

Quintor

Gradle (build tool):

- Dependencies (compatibility framework)
- Instellingen (SDK versie, applicatie versie)
- Verschillende releases (test, release)

Android Manifest

- Applicatie instellingen als icon en applicatie naam
- Definitie van al je schermen en welke opgestart moet worden als de app start
- Permissies

Java

- Alle java code en java tests

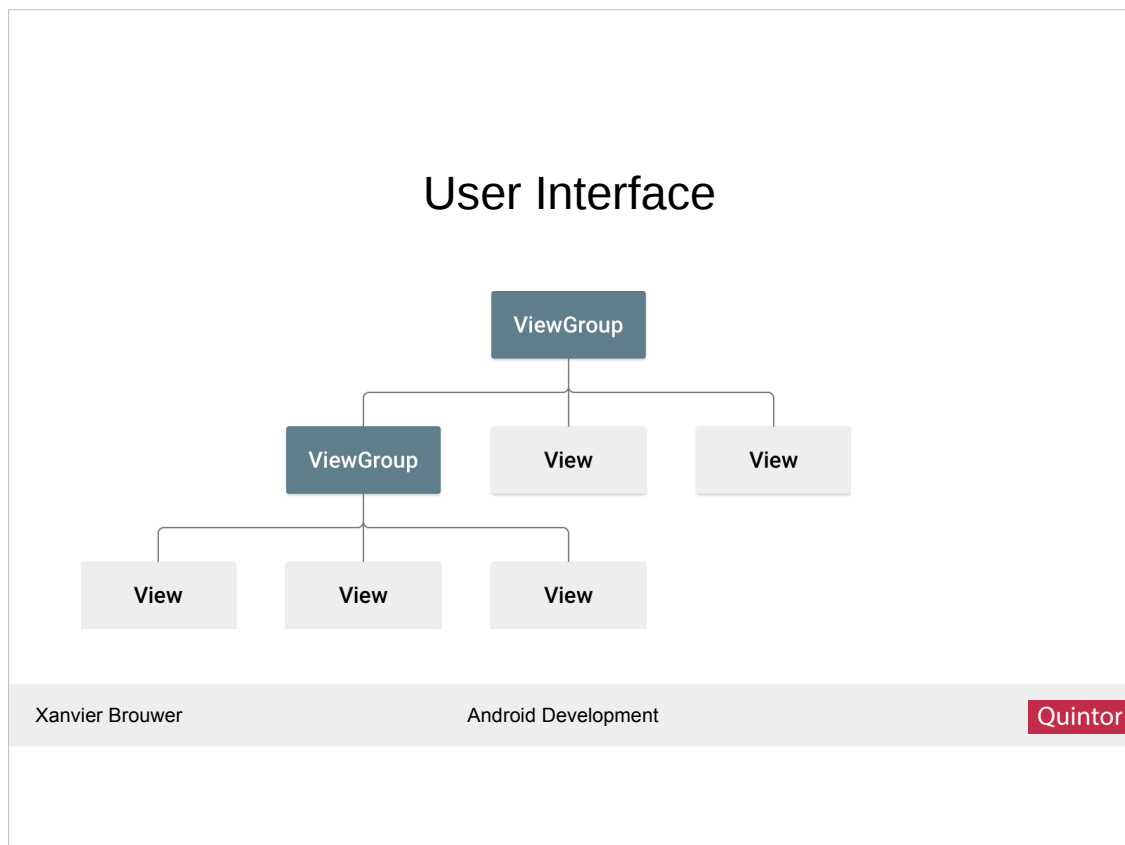
Res

- Alle XML bestanden met o.a. de definitie van je layouts.

Hands-on

Deel 1

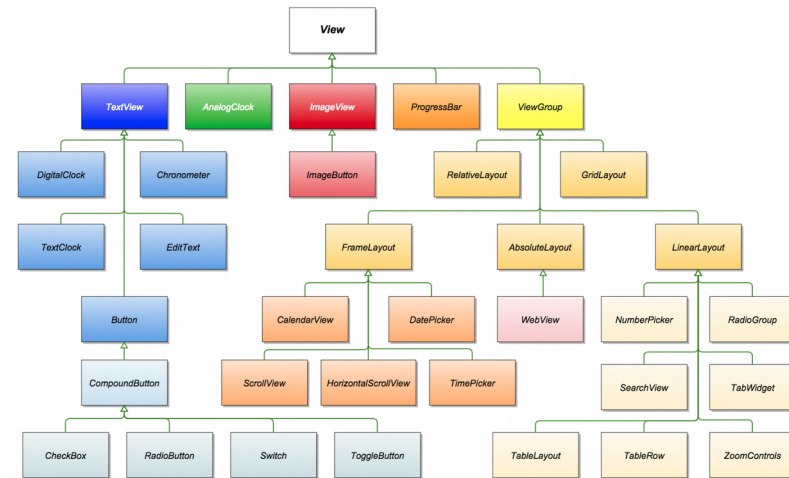
Ga naar https://github.com/xanvierb/android_handson en voer alles uit onder het kopje **Dependencies en eerste start-up**



Mobiele applicaties zijn erg visueel, het FE gedeelte is dus ook erg belangrijk.

De userinterface is een hiërarchische boom van View elementen in viewgroups.

View elements



Xanvier Brouwer

Android Development

Quintor

Alle elementen die je kunt zien of waar je iets mee kunt doen zijn View elements. Bijvoorbeeld TextViews, Buttons en plaatjes

Activity

- Scherm = Layout XML + Java code

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

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

Xanvier Brouwer

Android Development

Quintor

De viewgroups en viewelements tesamen kunnen een scherm vormen. Een scherm bevat alle viewgroup en view elements die gedefinieerd zijn in de XML code.

De functionaliteit zit in de java code.

Het lijkt hiermee op JavaFX, mocht iemand daar ervaring mee hebben.

Voorbeeld: een textView in een ConstraintLayout.

Hoe benader je Objecten die je hebt gedefinieerd in je XML code?

R-object

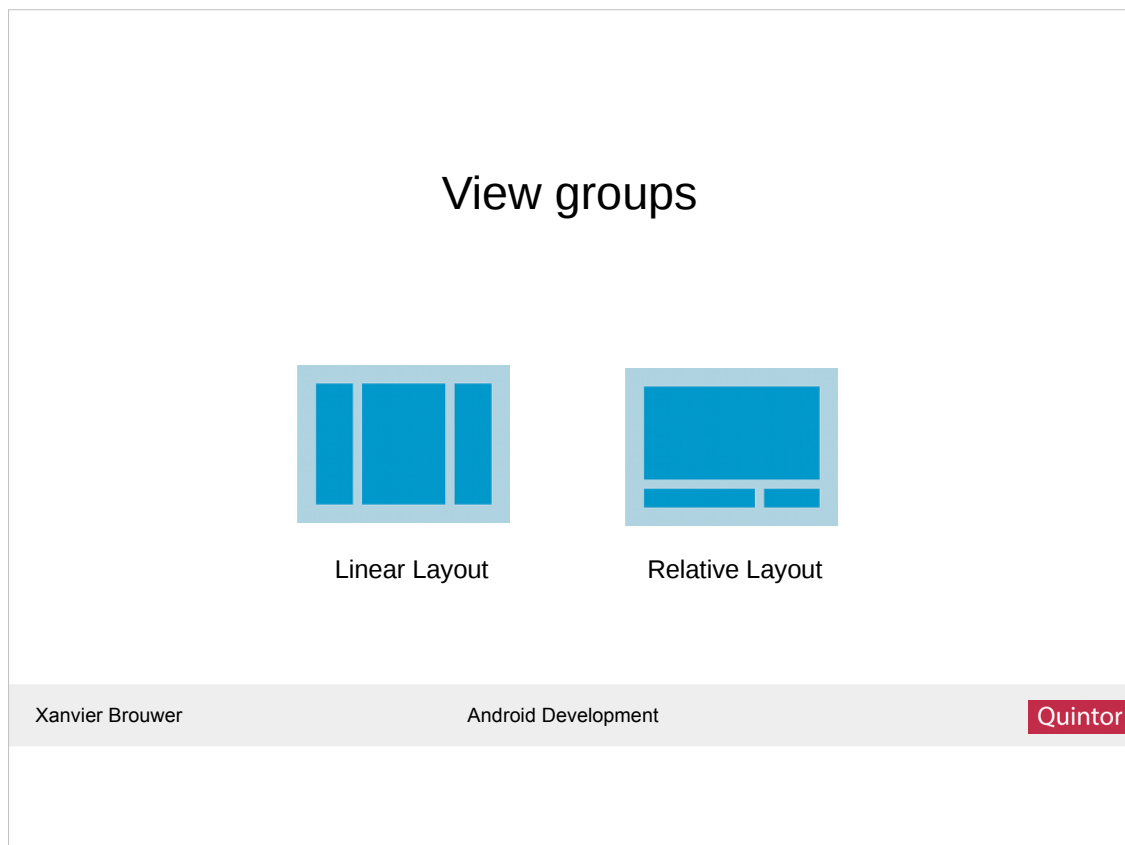
- Magic

```
TextView textDisplay =  
    (TextView) findViewById(R.id.text1);
```

Dit werkt magisch.

Je kunt met de statische klasse R alle in xml gedefinieerde elementen benaderen op hun ID. Dit wordt gemanaged door Android.

Soms moet opnieuw syncen.



Om view elements te ordenen kun je deze in ViewGroups plaatsen

2 belangrijkste layouts

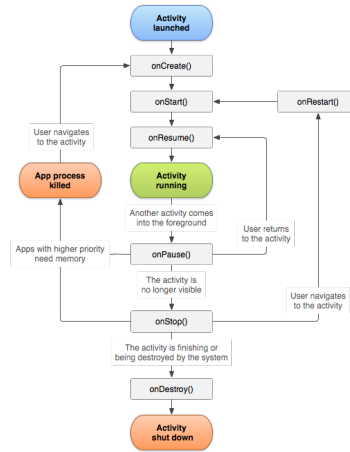
Linear Layout, waar alles naast elkaar komt te staan (of onder elkaar)

Relative layout, elementen relatief aan elkaar of relatief aan de parent.

Layouts kunnen in layouts

TableLayout, GridLayout, FrameLayout, AbsoluteLayout

Activity Lifecycle



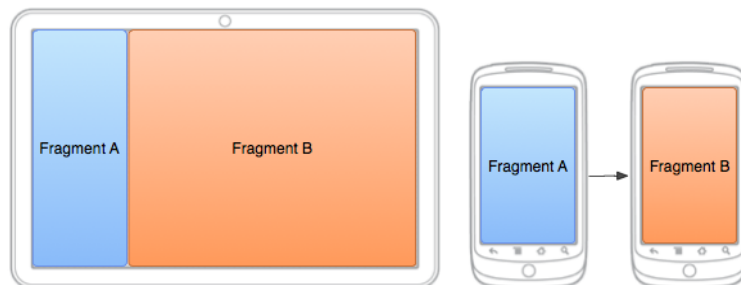
Xanvier Brouwer

Android Development

Quintor

doorlopen

Fragments



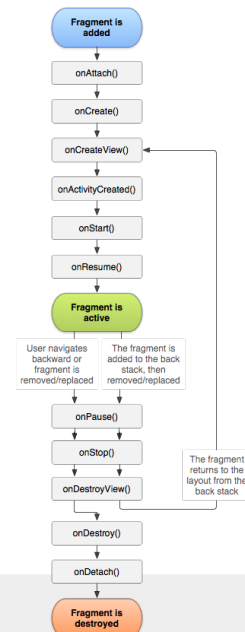
Xanvier Brouwer

Android Development

Quintor

Een activity kan bestaan uit fragments. Fragments zijn herbruikbare onderdelen en lijken in de opzet erg op Activities.
Je kunt ze gebruiken door op verschillende schermgroottes verschillende views weer te geven.

Fragment Lifecycle

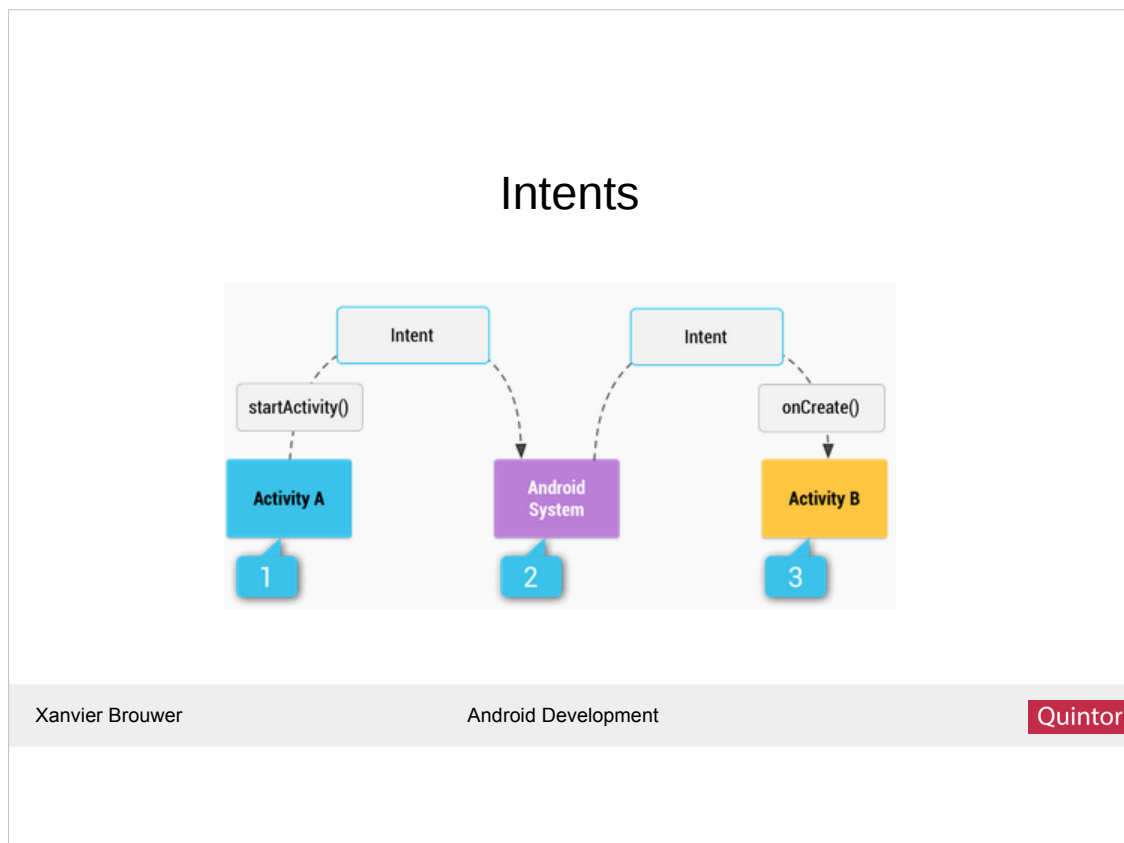


Xanvier Brouwer

Android Development

Quintor

doorlopen



Abstracte omschrijving van gedrag. Het bevat een actie en data.

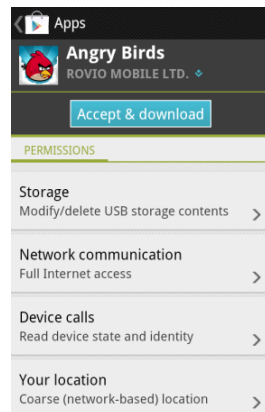
Wordt gebruikt om iets aan te sturen, bijvoorbeeld om naar een specifiek telefoonnummer te bellen. Of om alle contacten opgeslagen op de telefoon weer te geven om er een te kunnen selecteren.

Dit wordt ook gebruikt om van scherm A naar scherm B door te klikken.

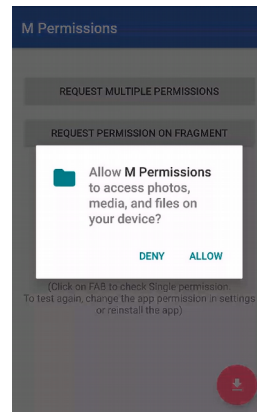
Voor vele intents heb je specifieke permissies nodig in de android manifest.

Er zijn vele Intents, zoek maar eens op.

Permissies



< API 23



>= API 23

Xanvier Brouwer

Android Development

Quintor

Voor API 23 werden alle permissies die aangegeven waren in de Android Manifest tijdens de installatie gevraagd.

Vanaf API 23 wordt dat gedaan in de applicatie zelf. Je moet hier wel zelf de code voor schrijven.

Als je meerdere API's ondersteund, moet je in de code zelf dus ook aangeven "If API > 22 vraag om permissies."

Hands-on

Deel 2

Ga naar https://github.com/xanvierb/android_handson en voer alles uit onder het kopje **Android applicatie structuur, User interface en Intents**

Resources

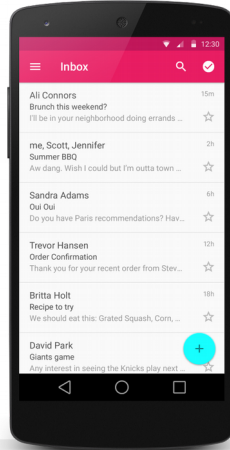
- String
- Drawable/mipmap
- Multi-language?

```
<TextView  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="@string/hello" />
```

Internationalization (i18n)

- `res/values/strings.xml`
- `res/values-fr/strings.xml`

ListViews



Xanvier Brouwer

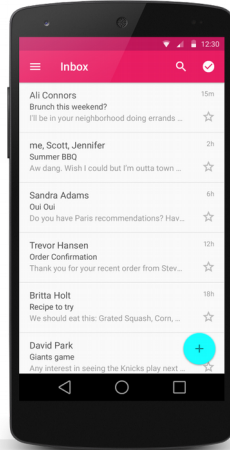
Android Development

Quintor

Op telefoons worden heel veel lijsten gebruikt. Erg handig om efficiënt data weer te geven.

Android heeft hier ListViews voor bedacht, maar heeft dit later vervangen voor RecyclerViews.

ListViews RecyclerView



Xanvier Brouwer

Android Development

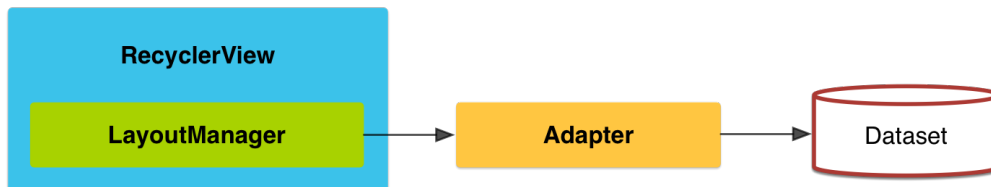
Quintor

RecyclerViews zijn een stuk complexer maar ook een stuk krachtiger. Zo kun je dynamisch de data aanpassen

Elementen die niet in beeld staan die worden niet geladen in het geheugen.

Je kunt ook dynamisch (makkelijker) elementen toevoegen en verwijderen.

RecyclerView



Hands-on

Deel 3

Ga naar https://github.com/xanvierb/android_handson en voer alles uit onder het kopje **Advanced User Interface**

Data persistentie

- SharedPreferences
- Schrijven naar bestand
- SQLite
 - Room

Xanvier Brouwer

Android Development

Quintor

Niet gebruiken tijdens de hands-on
SharedPreferences: key-value
SQLite: Structured data
Room: Laag over SQLite

Threads

- Code uitvoeren op verschillende (virtual) cores
- Applicatie niet blokkeren

Threading op Android

- UI thread
- Blocking
- Regels
 - Blokkeer de UI thread niet
 - Android UI niet benaderen vanaf andere dan de UI Thread
- Oplossing: Worker threads

Xanvier Brouwer

Android Development

Quintor

1 Main thread (UI thread)

Elk component draait in deze thread

Dit kan problemen opleveren voor intensief werk

De UI wordt geblokkeerd

Melding “Application is not responding” na 5 seconden

Een oplossing hiervoor zijn Worker Threads. Het nadeel hiervan is dat de complexiteit erg snel toeneemt. Vaak is het beter om gebruik te maken van...

AsyncTask

```
private class DownloadFilesTask extends AsyncTask<URL, Integer, Long> {  
    protected Long doInBackground(URL... urls) {  
        int count = urls.length;  
        long totalSize = 0;  
        for (int i = 0; i < count; i++) {  
            totalSize += Downloader.downloadFile(urls[i]);  
            publishProgress((int) ((i / (float) count) * 100));  
            // Escape early if cancel() is called  
            if (isCancelled()) break;  
        }  
        return totalSize;  
    }  
  
    protected void onProgressUpdate(Integer... progress) {  
        setProgressPercent(progress[0]);  
    }  
  
    protected void onPostExecute(Long result) {  
        showDialog("Downloaded " + result + " bytes");  
    }  
}
```

Xanvier Brouwer

Android Development

Quintor

...AsyncTasks

Moet worden ge-subclassed (extend AsyncTask)

Ideaal voor korte operaties, anders iets gebruiken uit concurrent package van java. (Executor, ThreadPoolExecutor, FutureTask)

REST

- HTTP
- Data ophalen van een API endpoint
- Dit is blocking

Xanvier Brouwer

Android Development

Quintor

REpresentational State Transfer

Regels hoe je HTTP protocol moet gebruiken

Je kunt hiermee data ophalen van een API endpoint

Dit is een blocking operation, hiervoor kun je goed een AsyncTask gebruiken.

Hands-on

Deel 4

Ga naar https://github.com/xanvierb/android_handson en voer alles uit onder het kopje **AsyncTask en REST calls**