



Androidskolen - Sesjon 1

“Developer” på 1-2-3

Hvem er vi?

Sondre Dyrkorn-Berg

Gjermund Karlsen Lie

Androidskolen

- Arrangeres over 5 kvelder
- Tre timers varighet (16-19)
- En time kan føres som interntid
- Legger vekt på praktiske øvelser
- Alle oppfordres til å lese hjemme

Overordnet målsetning

Alle deltakere skal gjennom Androidskolen få en god teoretisk forståelse av hva som kreves for å lage en vellykket applikasjon på Android plattformen. De skal gjennom øvelser få praktiske ferdigheter til å lage en applikasjon som inneholder de mest brukte komponentene og som følger best practice hva gjelder design og ytelse.

Androidskolen skal føre frem mot en vellykket "Associate Android Developer" sertifisering for samtlige deltakere.

Agenda

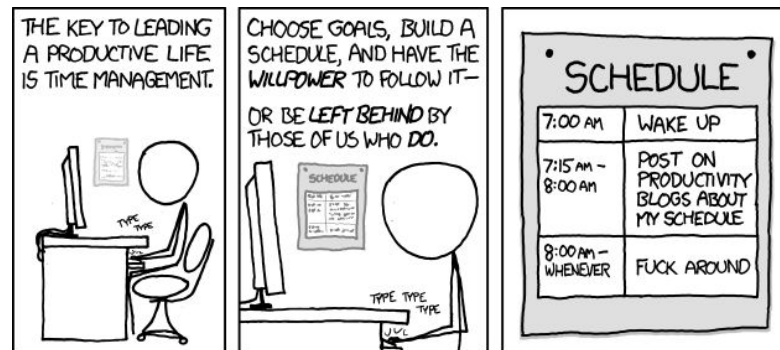
- Dagens målsetning
- Introduksjon til Android
- Praktisk del 1
- Androidprosjektets anatomi
- Androidutviklingens byggeklosser
- Praktisk del 2
- Manifest
- Permissions
- Praktisk del 3
- Oppsummering



Dagens målsetning

Etter denne sesjonen skal deltakerne kunne lage en enkel applikasjon ved hjelp av Android Studio samt at de skal kunne deployere denne applikasjonen på en emulator såvel som på en fysisk enhet.

Videre skal deltakerne ha en overordnet forståelse av de ulike hovedkomponentene en applikasjon er bygget opp av og en dypere forståelse av livssyklusen til en applikasjon. De skal også være kjent med hvordan man legger til logging i en applikasjon og hvordan man kan lese denne loggen.

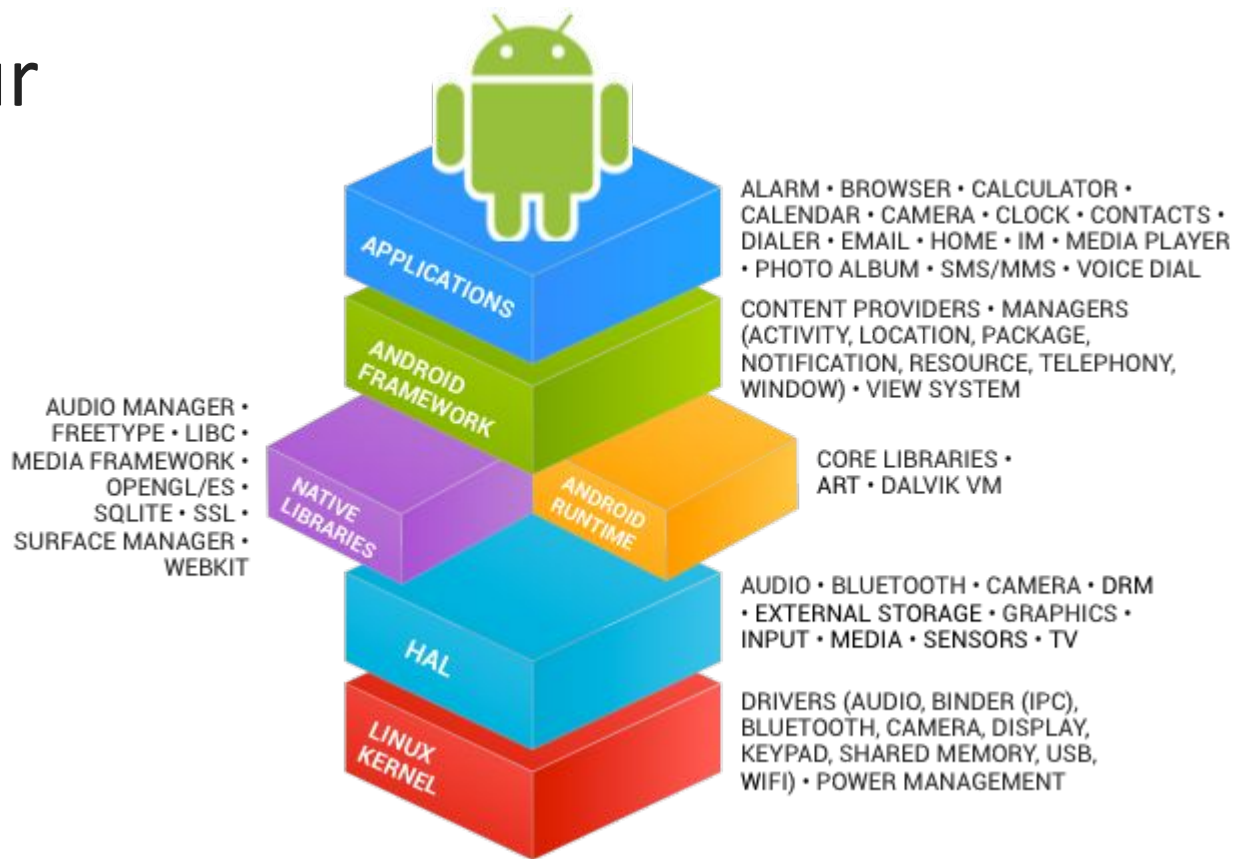


Praktisk del 1.

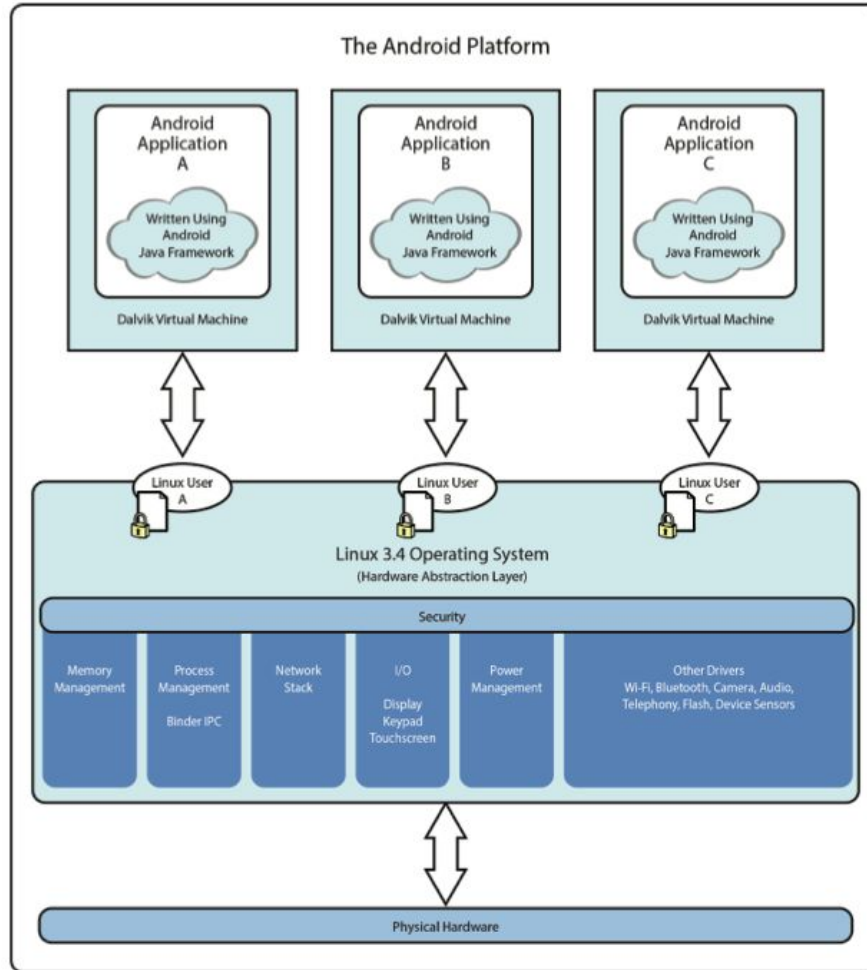
1. De som ikke har gjort det; installer Android Studio.
2. Installer Android SDK 6.0 (Marshmallow). Eller høyere! ;)
3. Opprett et nytt prosjekt, med Kotlin og en Basic Activity
4. Opprett en emulator som støtter Marshmallow.
5. Deploy til emulator.
6. Bli en utvikler på 1, 2, 3
7. Deploy til device.
8. Nyt resultatet og drikk kaffe.

I LIKE TO HAVE A
CUP OF COFFEE
TO RELAX AFTER
A LONG CUP OF
COFFEE

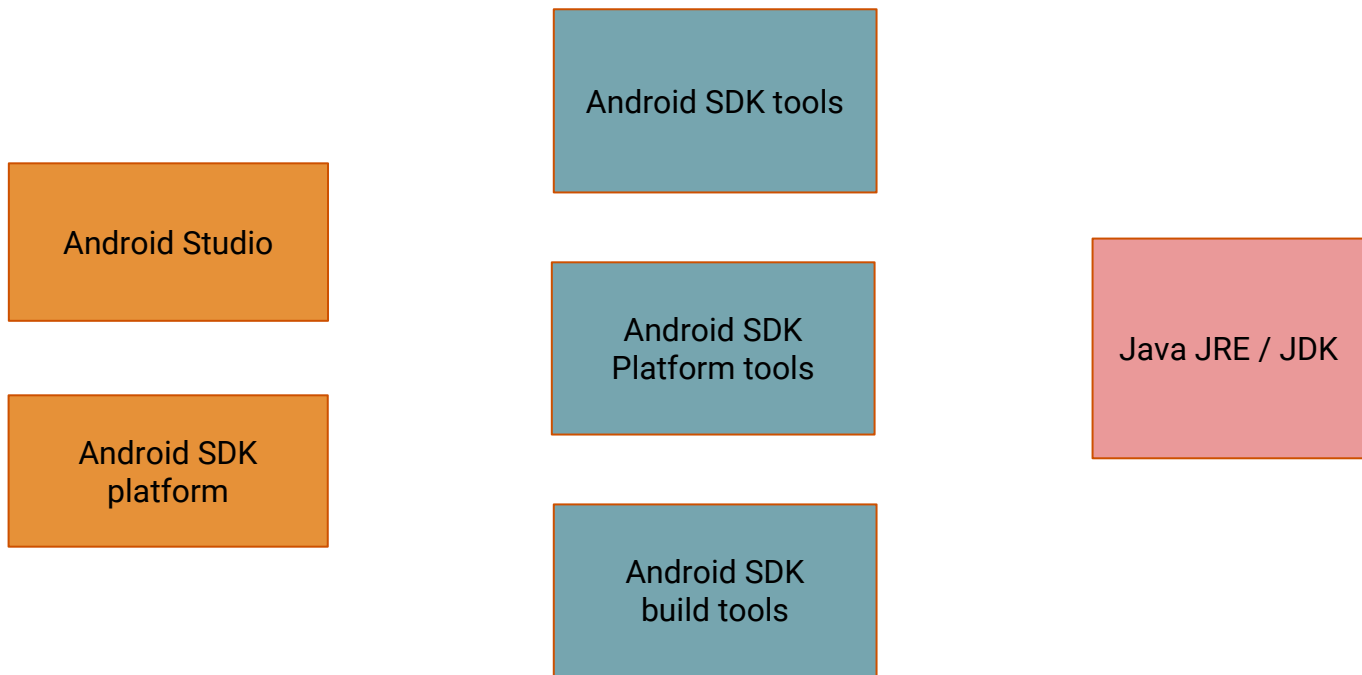
Arkitektur



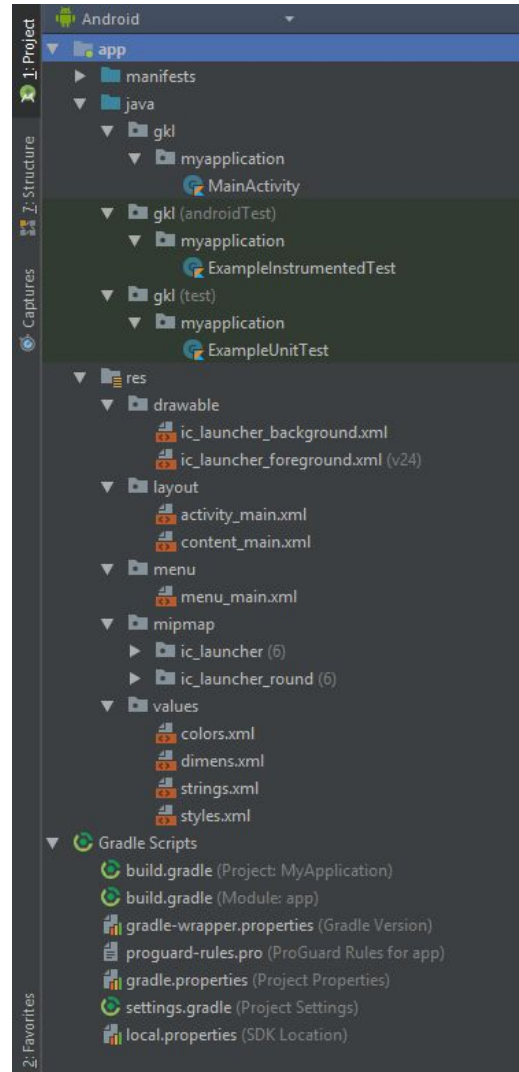
Arkitektur



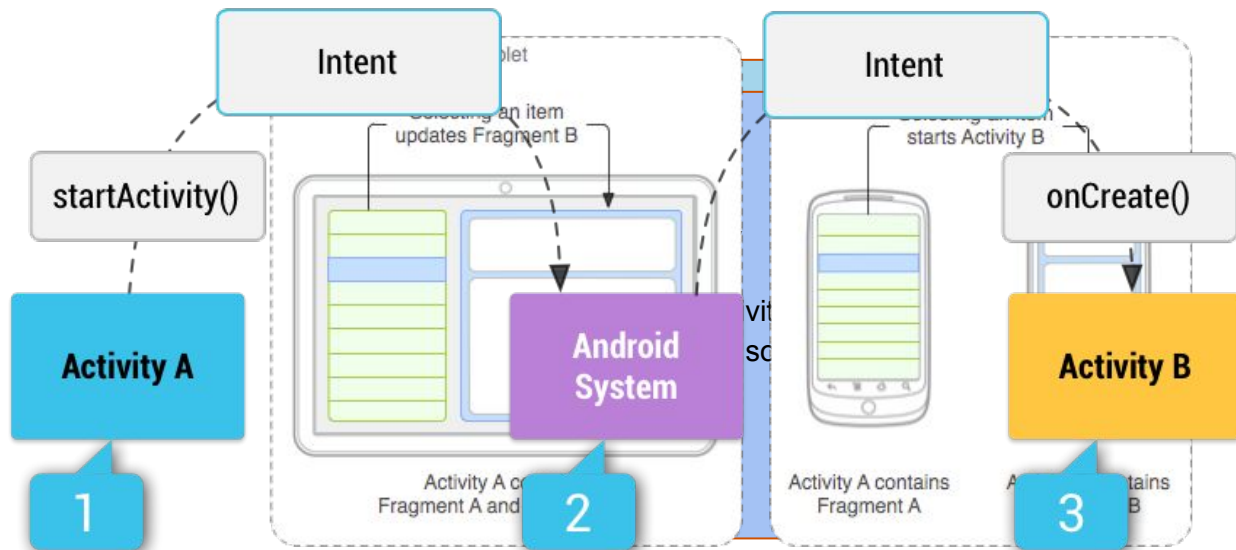
Hva trenger man for å tjene “one million dollars”



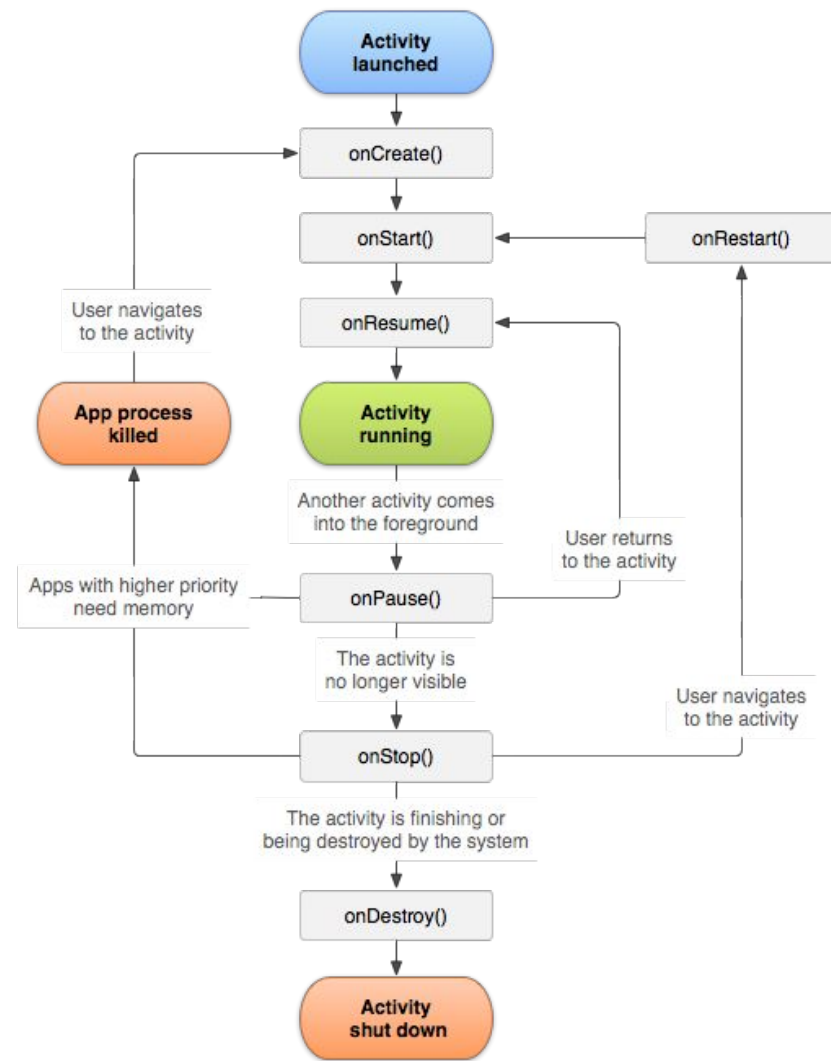
Androidprosjektets anatomi



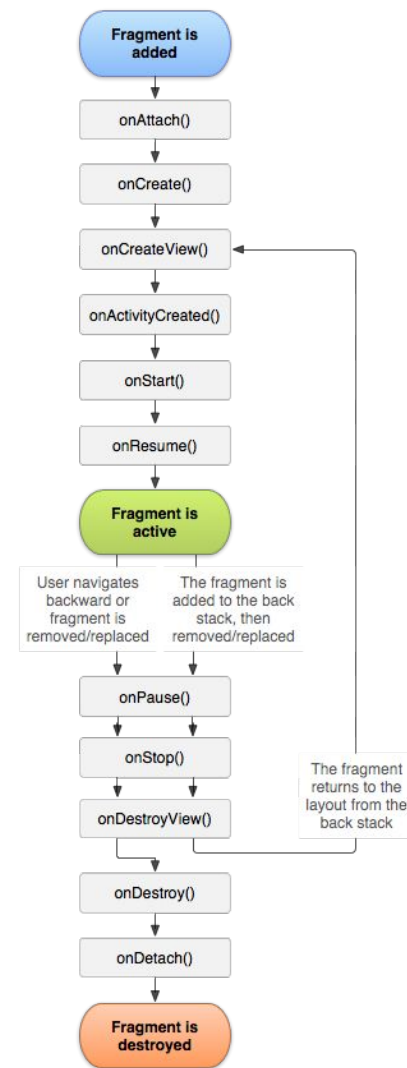
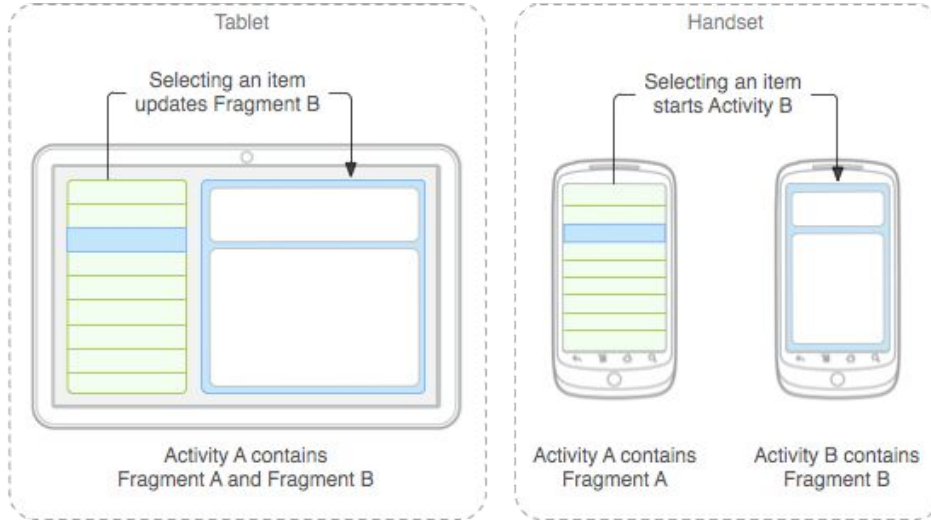
- Context
- Activity
- Fragment
- Intent
- Service
- Activity lifecycle
- Fragment lifecycle
- Logging



bouvét



Fragment lifecycle



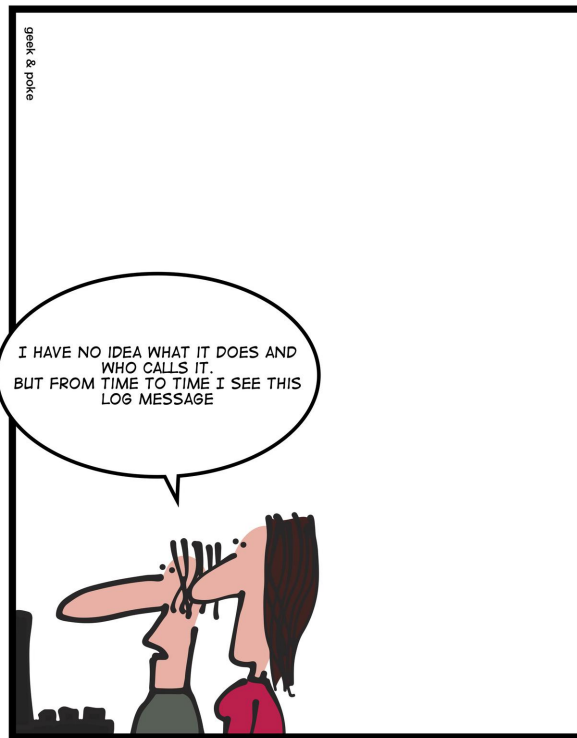
Logging

Logging natively blir gjort gjennom **android.util.Log**.

```
if (BuildConfig.DEBUG) {  
    Log.d("CATEGORY", messageAsString);  
}
```

“Logcat” har eget panel i Android Studio, med filtre

WHY LOGGING IS SO IMPORTANT



[INFO] com.doesntmatter.util.Etc - I'm here!

Praktisk del 2

1. Utvid appen vi opprettet tidligere
 - a. Override alle lifecycle callbackene i aktiviteten (MainActivity)
 - b. Legg inn et log statement i hver lifecycle callback.
2. Deploy app på fysisk device og se om man klarer å trigge alle lifecycle eventene.

Manifest

Gir nødvendig informasjon om appen til android systemet som f.eks:

- Rettigheter som trengs.
- Komponenter den inneholder.
- Hva slags hardware den er avhengig av.
- Meta-data som api keys

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

    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the 'MyLocation' functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">

        <!--
            The API key for Google Maps-based APIs is defined as a string resource.
            (See the file "res/values/google_maps_api.xml").
            Note that the API key is linked to the encryption key used to sign the APK.
            You need a different API key for each encryption key, including the release key that is used to
            sign the APK for publishing.
            You can define the keys for the debug and release targets in src/debug/ and src/release/.
        -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key"/>

        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

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

</manifest>
```

Permissions

Man spør om permissions:

1. Manifest:
`<uses-permission android:name="android.permission.READ_CONTACTS"`
2. Runtime og Up front.
 - a. Normal permissions blir gitt automatisk, mens dangerous permissions må resolves runtime.
 - i. Sjekk for om man har permissions før bruk
 - ii. Sjekk om man bør gi en forklaring
 1. Dersom ja - Vis noe til brukeren asynkront. Når brukeren har sagt ok, spør om permission.
 - iii. Spør kun om spesifikk permission.
3. Man kan definere sine egne permissions!

Praktisk del 3

1. Last ned koden for AndroidSkolen fra Bouvets GitHub
 - a. <https://github.com/bouvét/AndroidSkolen.git>
 - b. Åpne prosjektet Sesjon1DisplayingBitmaps
2. Legg inn nytt hurtigstartikon (mipmap)
3. Fortell google play at appen kun skal vises for enheter med kamera.
4. Det er et par ting som må fikses:
 - a. LoginActivity laster ikke GUI. Den skal laste activity_login.xml.
 - b. Login knappen starter ikke ImageGridActivity.
 - c. Appen mangler permission for å skrive til external storage.
5. ImageGridActivity skal logge brukernavn og passord som ble lagt inn i LoginActivity.

Oppsummering

- Vi utvikler på en Linux platform utvidet med rammeverksbiblioteker
- Activity, fragment, service, intent
- Lifecycle
- Manifest
- Permissions
- Logging

Quiz

I hvilke(n) klasse(r) implementerer man onCreate() og hva skal man gjøre her?

Hva gjør metoden setContentView()?

Hvordan får man en referanse til et view i aktiviteten?

Hvordan logger man?

Kan man sende med data når man starter en annen aktivitet?

Til neste gang