

Testing

Why should I test code?

- Reduce bugs in new features
- Reduce bugs in existing features
- Reduce cost of change
- Allow refactoring
- Reduce fear
- Make development faster

Why should I test my android app?

- A lot of special cases (orientation change, no connectivity, ...)
- Run tests on different devices

Unit Testing using jUnit

Used for everything that has nothing to do with android

- jUnit

Instrumentation Testing

Used for everything you cannot or do not want to remove dependencies

- Android Instrumentation
- Robotium
- Espresso

→ Needs a device or an emulator to run!

Easy to use library to write readable Android UI Tests

```
@Test public void greeterSaysHello() {  
    onView(withId(R.id.name_field)).perform(typeText("Steve"));  
    onView(withId(R.id.greet_button)).perform(click());  
    onView(withText("Hello Steve!")).check(matches(isDisplayed()));  
}
```

Select a view you want to interact with using onView

```
// Use a single identifier  
onView(withId(R.id.buttonAdd))  
  
// Use multiple identifiers  
onView(allOf(withId(R.id.counterValue), withText("0")))
```

Perform actions on the views using perform(...)

```
onView(withId(R.id.buttonAdd)).perform(click());
```

Perform checks on the given view

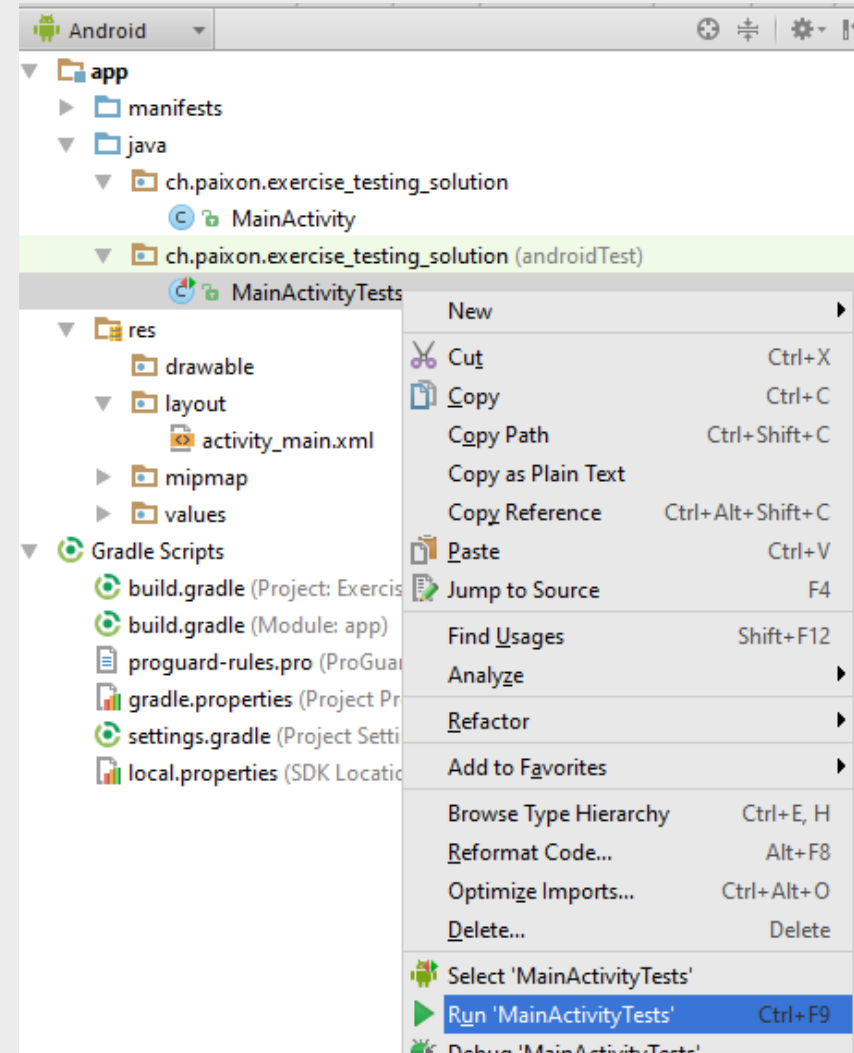
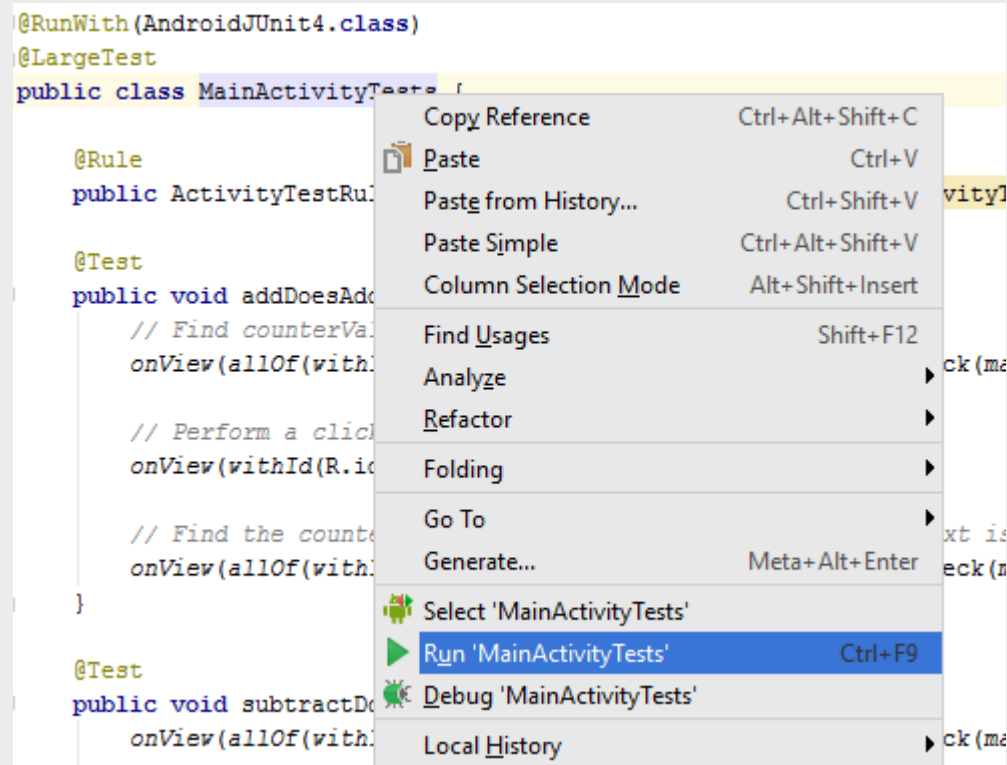
```
onView(withId(R.id.counterValue)).check(matches(withText("1")));
```

Espresso Cheatsheet available here:

<https://developer.android.com/training/testing/espresso/cheat-sheet>

Testing – Android Studio

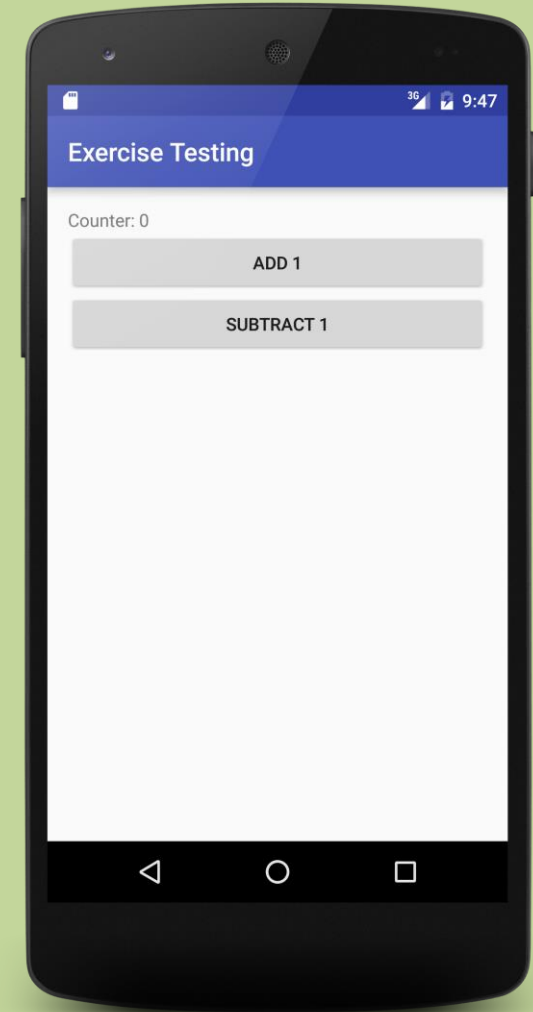
Paixon



Aufgaben

1. Analyze and run the test for the Add-method
2. Add a test for the subtract method

Projekt: Exercise_Testing



Problem

Test hat sehr viel Know-How über das UI

Lösung

Abstraktion einzelner "Pages" in ein Page-Object

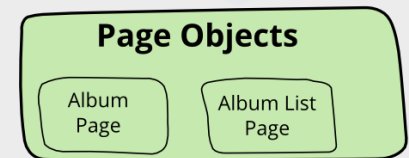
➔ Stellt Funktionalität zur Verfügung

Mehr Infos dazu:

<https://martinfowler.com/bliki/PageObject.html>

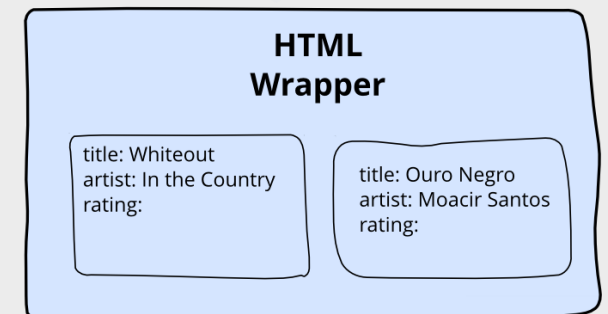
this API is about
the application

```
selectAlbumWithTitle()  
getArtist()  
updateRating(5)
```



this API is
about HTML

```
findElementsWithClass('album')  
findElementsWithClass('title-field')  
getText()  
click()  
findElementsWithClass('ratings-field')  
setText(5)
```



Google Play Services

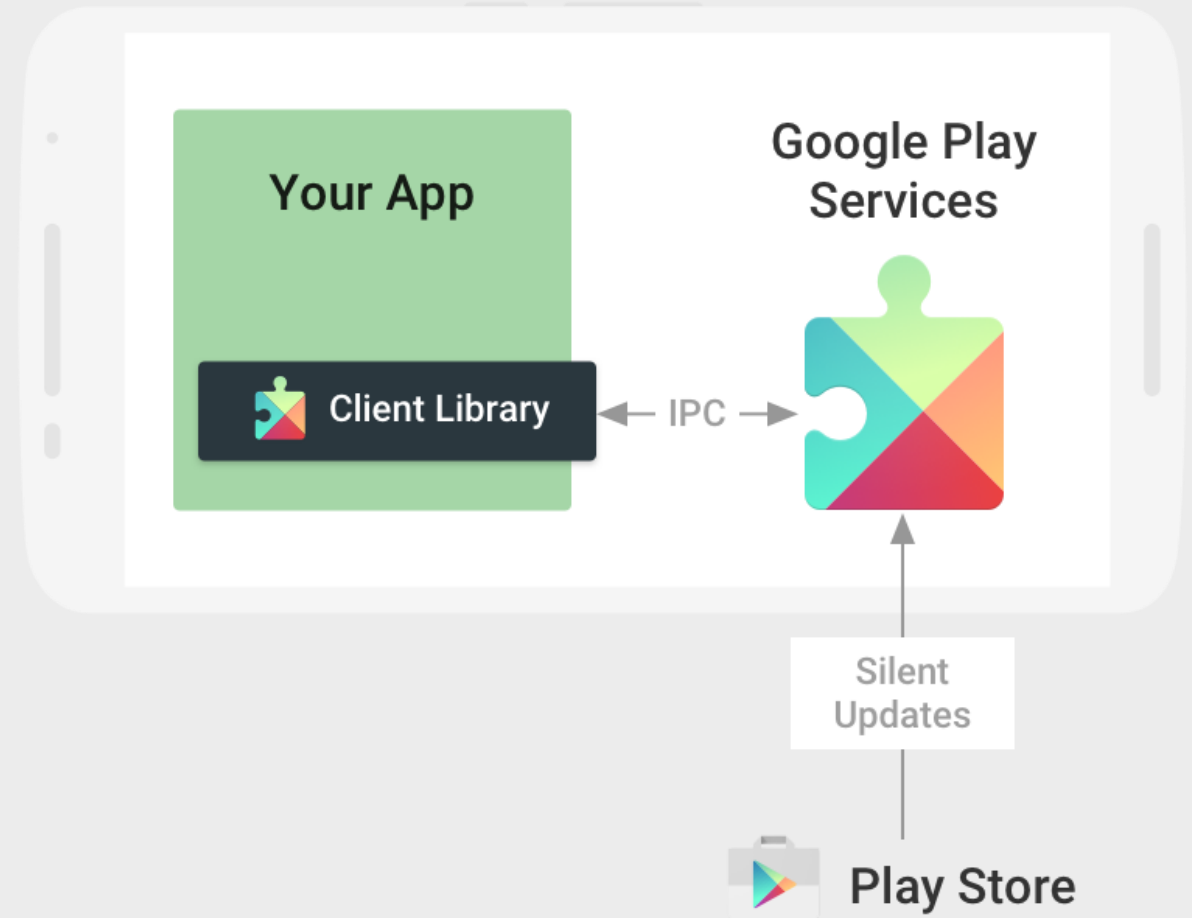
Background service and API to access Google Services

- Google Play Game Services
- **Location APIs**
- Google+
- **Google Maps**
- Google Drive
- Cast
- Ads
- Wallet
- Google Fit
- Google Analytics
- ...

- High availability
- Silent update
- Shared accross multiple apps
- Access Google Play Services through Client Library

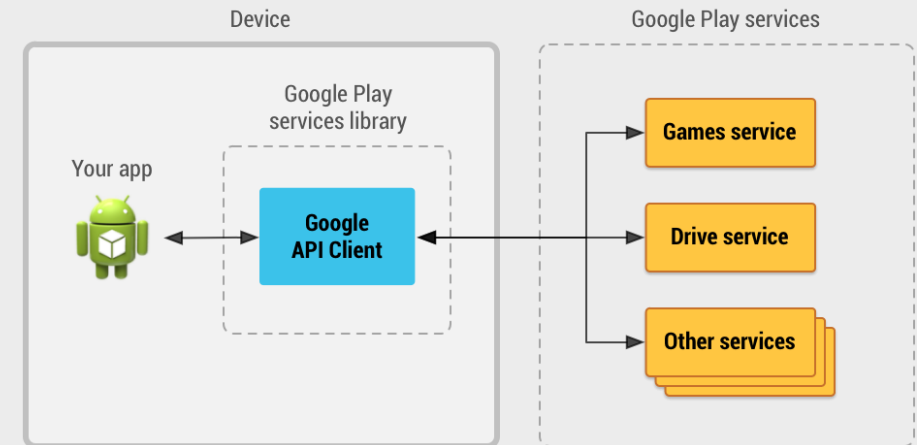
Minimum Requirements

- Android 2.3
- Emulator 4.2.2



Create GoogleApiClient instance

```
client = new GoogleApiClient.Builder(this)
    .addConnectionCallbacks(this)
    .addOnConnectionFailedListener(this)
    .addApi(LocationServices.API)
    .addApi(...)
    .build();
```



Implement Callback-Interfaces

GoogleApiClient.ConnectionCallbacks

GoogleApiClient.OnConnectionFailedListener

Connect the Service Client

```
@Override  
protected void onResume() {  
    super.onResume();  
  
    client.connect();  
}
```

Disconnect the Service Client

```
@Override  
protected void onPause() {  
    super.onPause();  
  
    client.disconnect();  
}
```

Location APIs

Get the User Location

Where your phone has your location from

- GPS `android.permission.ACCESS_FINE_LOCATION`
- Network/WiFi `android.permission.ACCESS_COARSE_LOCATION`

Fused Location Provider

Access to location depending on given permissions.

```
FusedLocationProviderClient client =  
LocationServices.getFusedLocationProviderClient(this);
```


Get the last known location

```
client.getLastLocation().addOnSuccessListener(new OnSuccessListener<Location>() {  
    @Override  
    public void onSuccess(Location location) {  
        // Use the location  
    }  
});
```

- Accuracy depending on the available permission

Provide a LocationCallback

```
private LocationCallback locationCallback = new LocationCallback() {  
    @Override  
    public void onLocationResult(LocationResult locationResult) {  
        // Use locationResult.getLocations() -> Location  
    }  
};
```

Create a LocationRequest

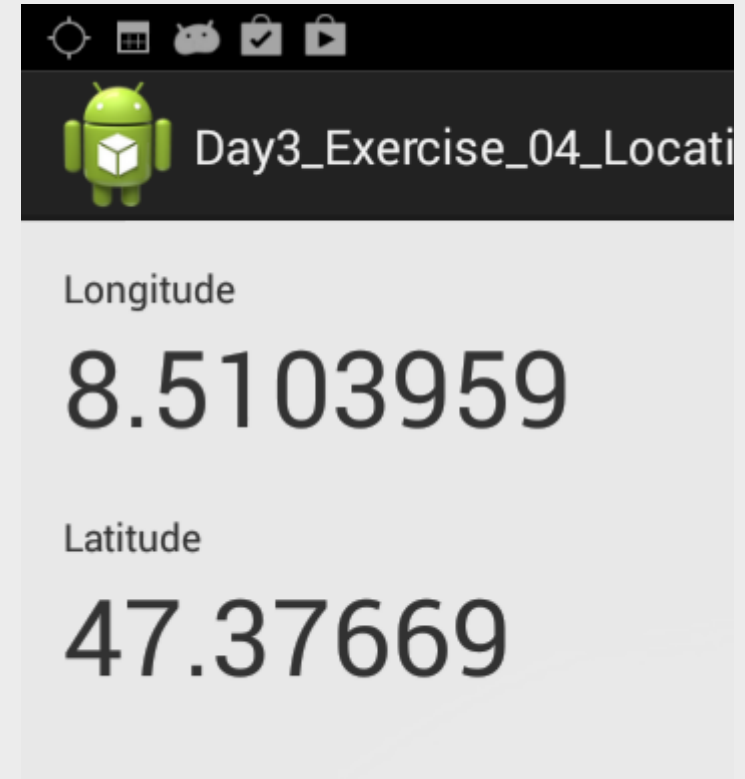
```
LocationRequest locationRequest = new LocationRequest();  
locationRequest.setInterval(10000);  
locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
```

Register for updates

```
client.requestLocationUpdates(locationRequest, locationCallback, null);
```

How useful is it to present Longitude/Latitude to your User?

So we need the address which does represents this two numbers...



Reverse Geocoding

Vorhanden: Longitude/Latitude

Gesucht: Genaue Adresse

Geocoding

Vorhanden: Genaue Adresse

Gesucht: Longitude/Latitude

Android Framework does provide a class «Geocoder»

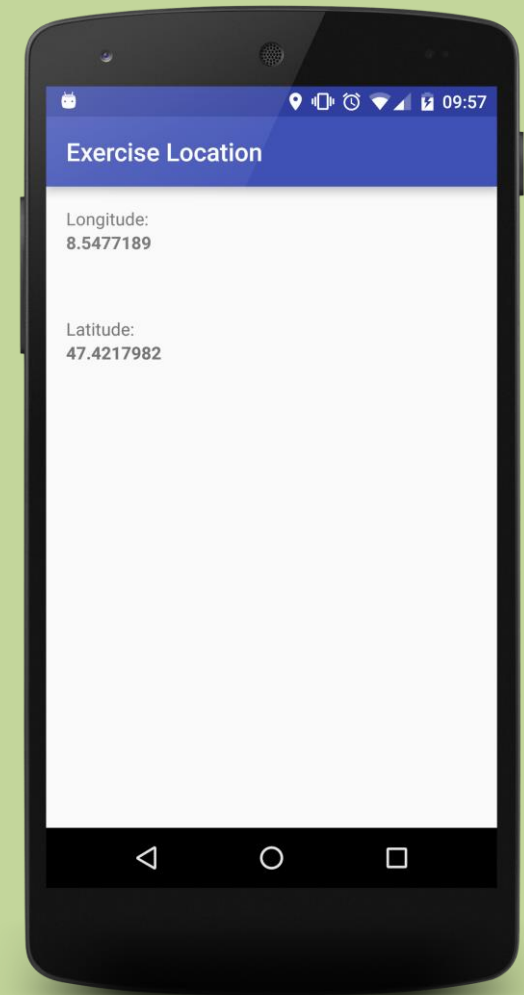
- `getFromLocation(longitude, latitude, count)`
- `getFromLocation(name, count)`

Hint: Use `Geocoder.isPresent()` to check whether the service is available!

Aufgaben

Zeige die aktuelle Position (Longitude/Latitude) des Users auf dem UI an.

Projekt: Exercise_Location



Location APIs

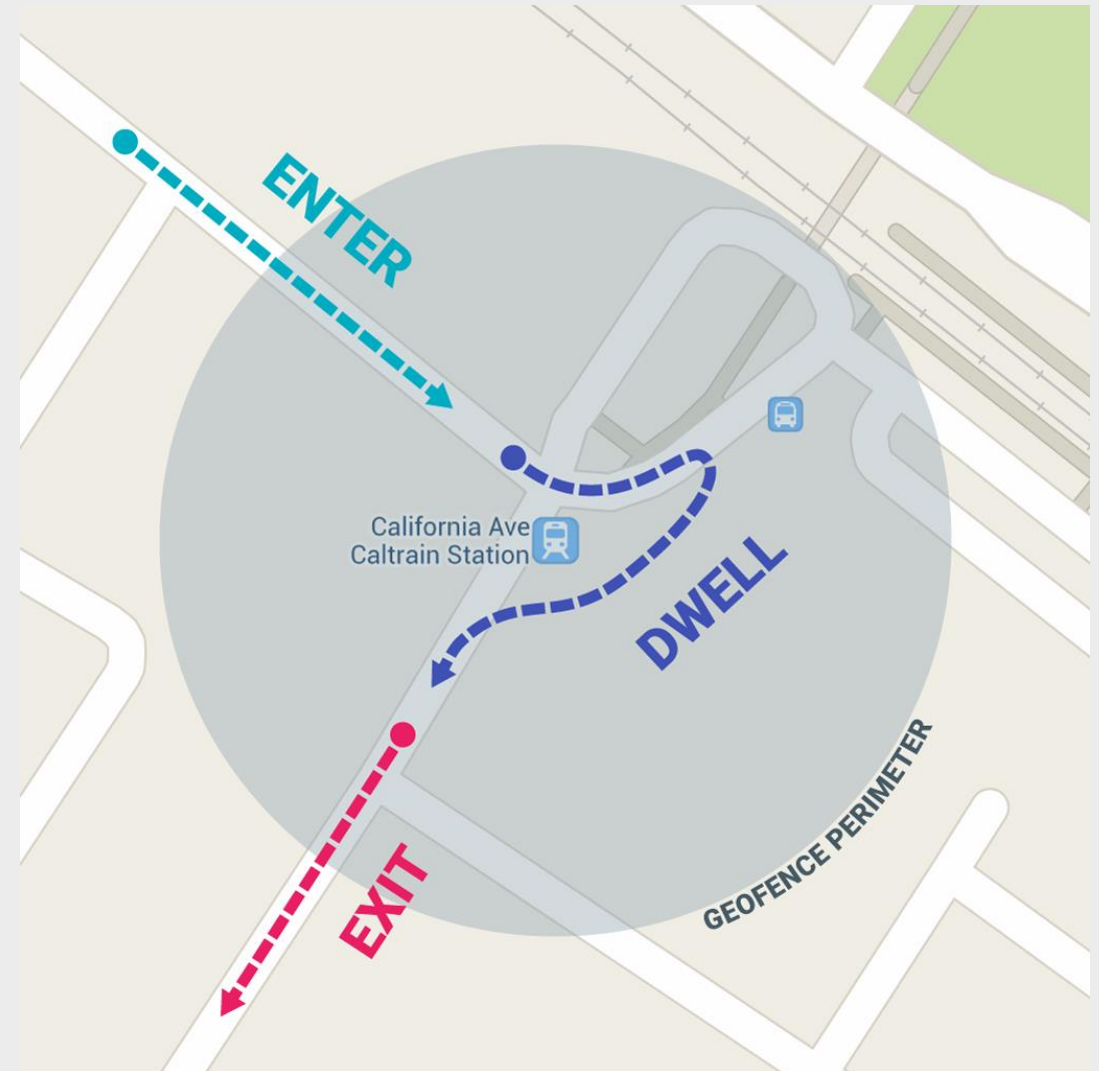
Geofencing

Define points of interests

- Longitude
- Latitude
- Radius

Get notified when the user...

- ... enters your POI
- ... is within your POI
- ... exits your POI



Location APIs

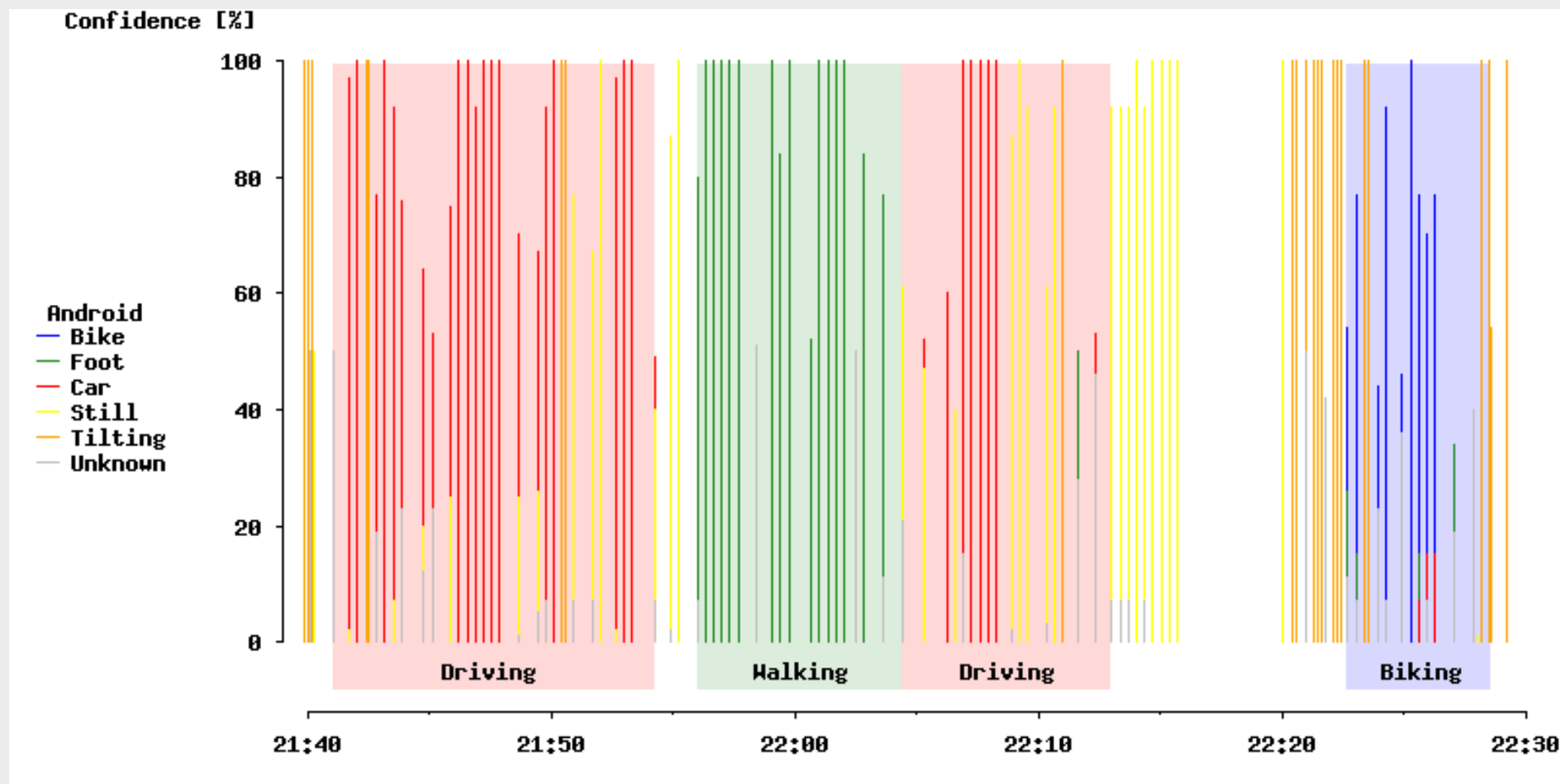
Activity recognition

Detect user current physical activity

- Walking, driving, standing still

How does it work?

- Android periodically waking up the device
- Low power sensors used
- Update interval depending on defined update interval



Networking

Achtung:

Sämtliche Netzwerk-Operationen müssen in jedem Fall in einem Hintergrund-Thread ausgeführt werden.

Android bietet HttpURLConnection und HttpsURLConnection

```
URL url = new URL("http://paixon.ch");
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
try {
    InputStream in = new BufferedInputStream(urlConnection.getInputStream());
    readStream(in);
} finally {
    urlConnection.disconnect();
}
```

- Raw Daten → Objekt
- Async-Issues
- Cancel von Aufrufen
- Wiederholende/Ersetzende Aufrufe
- Priorisierung
- Retry-Mechanismen
- Effizienz (Gzip, HTTP2, ...)

Es gibt Libraries die euer Leben vereinfachen

- OkHTTP
- Retrofit
- Volley

Retrofit

“A type-safe HTTP client for Android and Java”

- Open-Source Projekt entwickelt von Square
- Vom JSON-API zum typisierten Service
- Baisert auf OkHTTP (GZip, Caching, HTTP2, ...)

<http://square.github.io/retrofit/>

Schritt 1: Resource analysieren

Resource URL

<http://transport.opendata.ch/v1/connections>

----- Base URL-----/---- Path ----

Request-Parameters

Request Parameters			
Name	Required	Description	Example
from	required	Specifies the departure location of the connection	Lausanne
to	required	Specifies the arrival location of the connection	Genève
via	optional	Specifies up to five via locations. When specifying several vias, array notation (<code>via[]=via1&via[]=via2</code>) is required.	Bern
date	optional	Date of the connection, in the format YYYY-MM-DD	2012-03-25

Schritt 2: Data Transfer Object erstellen

```
public class ConnectionDto {  
  
    public String from;  
    public Date fromTime;  
  
    public ArrivalLocationDto arrivalLocationInfos;  
    public List<LocationDto> stopOvers;  
}
```

- Aufzählungen werden als List<>'s dargestellt
- Verschachtelungen für Objekte

Schritt 3: Service-Interface für die Resource erstellen

```
public interface ConnectionService {  
  
    @GET("connections")  
    Call<ConnectionDto> searchConnections(@Query("from") String from, @Query("to") String to);  
  
}
```

Aufruf an:

<BASEURL>/connections?from={String from}&to={String to}

Schritt 4: Bootstrap Retrofit

```
Retrofit retrofit = new Retrofit.Builder()  
    .baseUrl("http://transport.opendata.ch/v1/")  
    .addConverterFactory(GsonConverterFactory.create())  
    .build();
```

Schritt 5: Service intanzieren

```
ConnectionService service = retrofit.create(ConnectionService.class);
```

Schritt 6: Call absetzen

```
service.searchConnections("Zürich", "Bern").enqueue(new Callback<ConnectionDto>() {  
    @Override  
    public void onResponse(Call<ConnectionDto> call, Response<ConnectionDto> response) {  
        if (response.isSuccessful()) {  
            ConnectionDto connection = response.body();  
            // Handle result...  
        }  
    }  
  
    @Override  
    public void onFailure(Call<ConnectionDto> call, Throwable t) {  
    }  
});
```

Async Call → Handler on Main Thread

Aufgaben

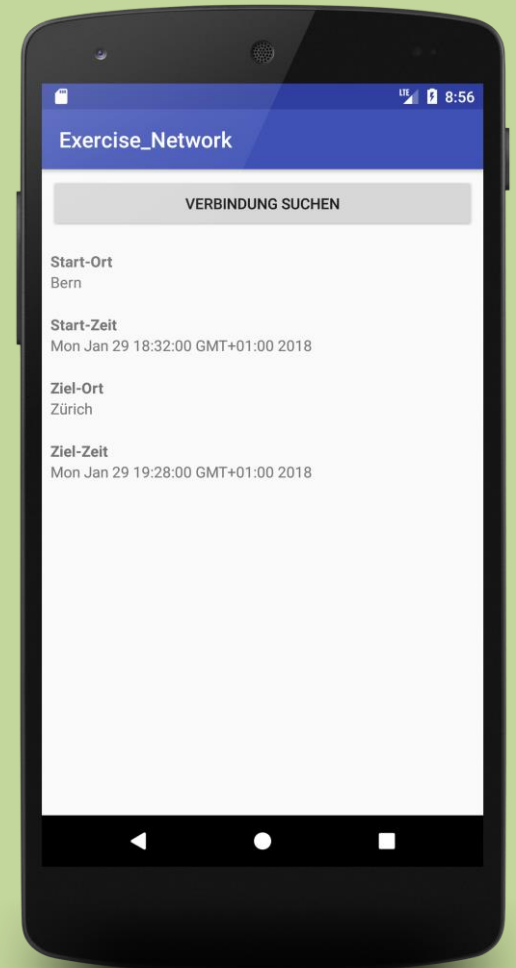
Part 1:

Zeige die nächste Abfahrts und Ankunfts-Zeit der Verbindung Zürich
→ Bern an

Part 2:

Überlege dir, welche Informationen du für deine App benötigst.
Baue die DTO's entsprechend auf.

Projekt: Exercise_Network

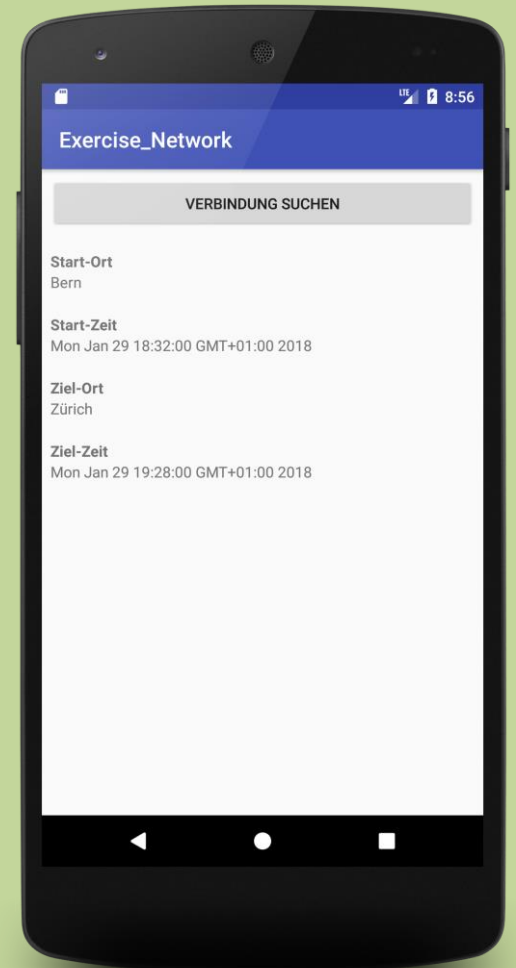


App-Architektur

Aufgaben

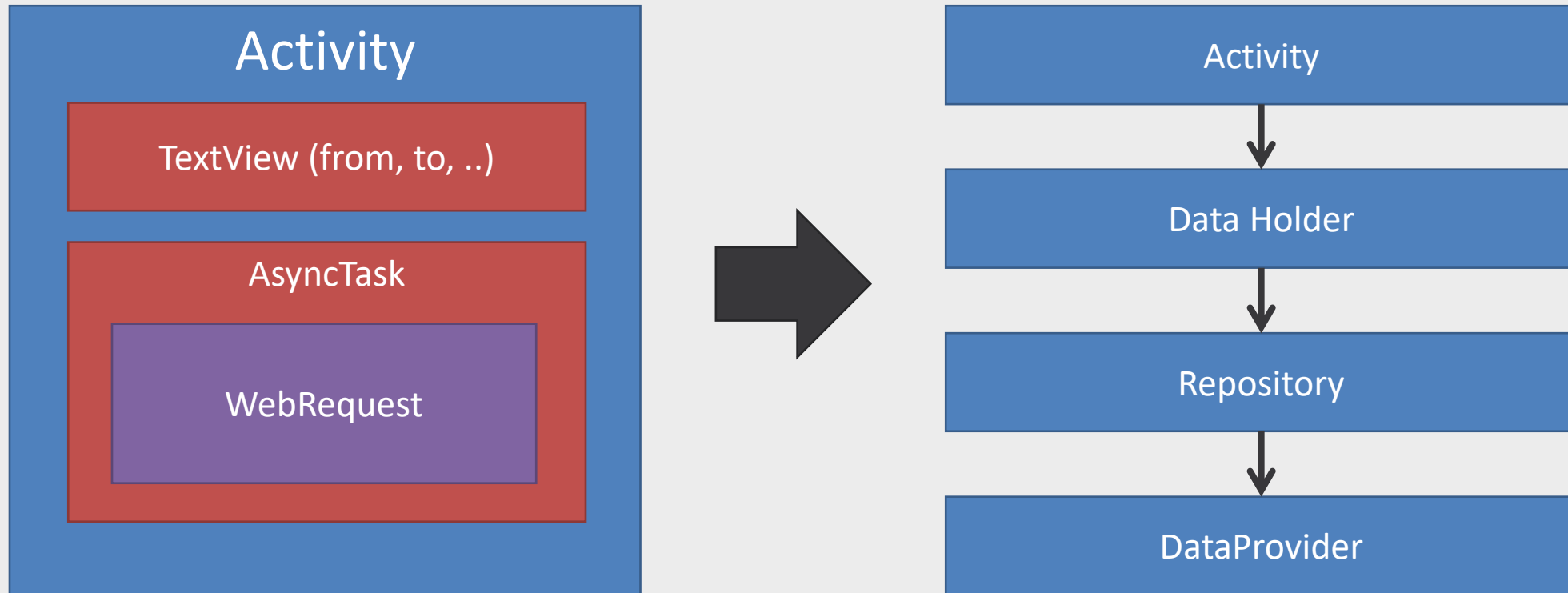
Wo seht ihr die Probleme bei der Lösung der Networking-Übung?

Denkt auch an die alten Übungen zurück – Was könnte in einem grösseren Projekt zu Problemen führen?



Probleme

- Prinzip der Single-Responsibility
 - Testbarkeit der Activity
 - Wiederverwendbarkeit
 - Android Abhängigkeit
- State wird nicht wiederhergestellt
- Manuelles registrieren von Callbacks (z.B. Location Listener)

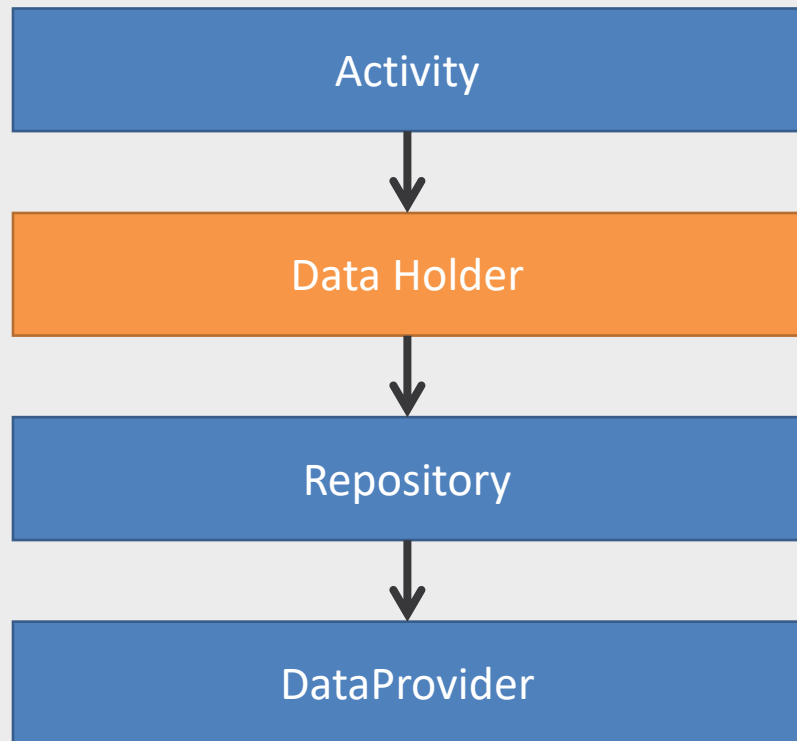


- Android war früher ein “Architektur-Freier-Raum”
- Referenz-Architektur über “Android Architectural Components”
- Eingebunden als externe Library

```
// ViewModel and LiveData
implementation "android.arch.lifecycle:extensions:1.1.0"
// alternatively, just ViewModel
implementation "android.arch.lifecycle:viewmodel:1.1.0"
// alternatively, just LiveData
implementation "android.arch.lifecycle:livedata:1.1.0"

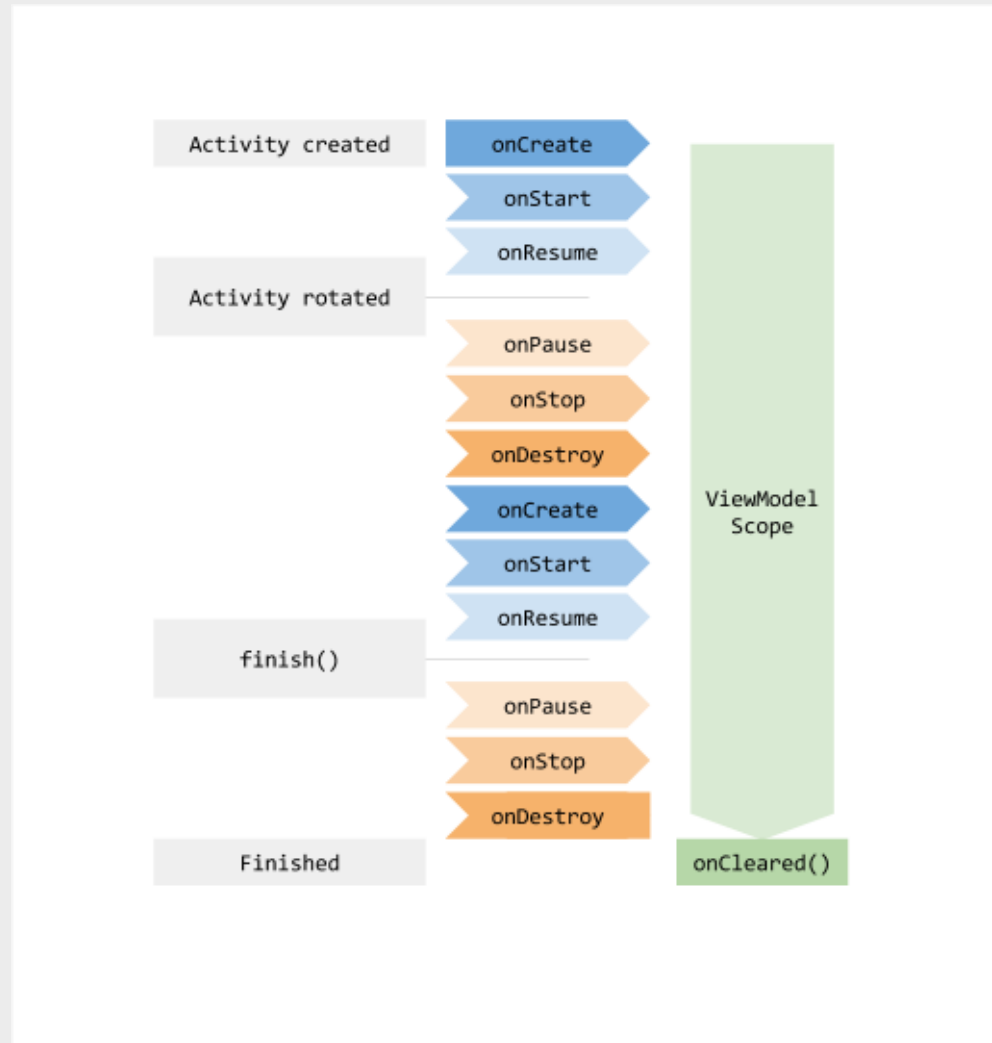
annotationProcessor "android.arch.lifecycle:compiler:1.1.0"
```

<https://developer.android.com/topic/libraries/architecture/adding-components.html>



ViewModel

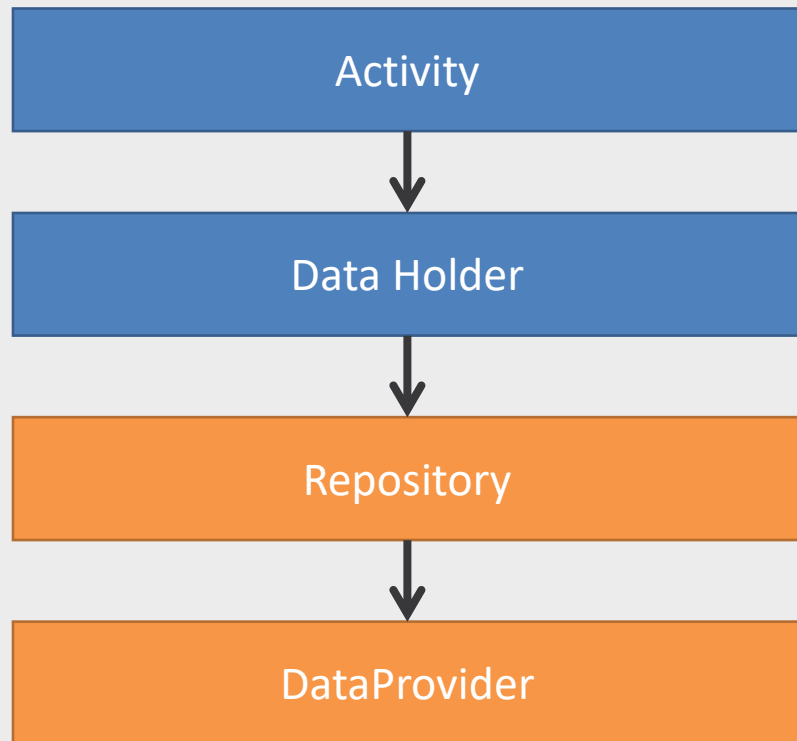
- Speichert UI-Daten
- Überlebt ConfigurationChange
- Nie eine Referenz auf das UI
- Liefert Daten über LiveData-Objekte
 - Observable Dataholder
 - Lifecycle Aware



```
public class MyActivity extends AppCompatActivity {  
    public void onCreate(Bundle savedInstanceState) {  
        MyViewModel model = ViewModelProviders.of(this).get(MyViewModel.class);  
        model.getUsers().observe(this, users -> {  
            // update UI  
        });  
    }  
}
```

```
public class MyViewModel extends ViewModel {
    private MutableLiveData<List<User>> users;
    public LiveData<List<User>> getUsers() {
        if (users == null) {
            users = new MutableLiveData<List<Users>>();
            loadUsers();
        }
        return users;
    }

    private void loadUsers() {
        // Do an asynchronous operation to fetch users.
        users.setValue(...);
    }
}
```



Repository

- Abstraktion der DataProvider

DataProvider

- Führt den Datenzugriff aus
 - Web
 - Datenbank
 - Filesystem
 - ...

DEMO

Wie weiter...

Paixon

Dependency Injection

Dagger (<https://google.github.io/dagger/>)

Data Binding

<https://developer.android.com/topic/libraries/data-binding/index.html>

UI (Activity, Fragments)

Android Instrumentation Tests → Espresso

ViewModel

Mock Repository → Unit-Tests

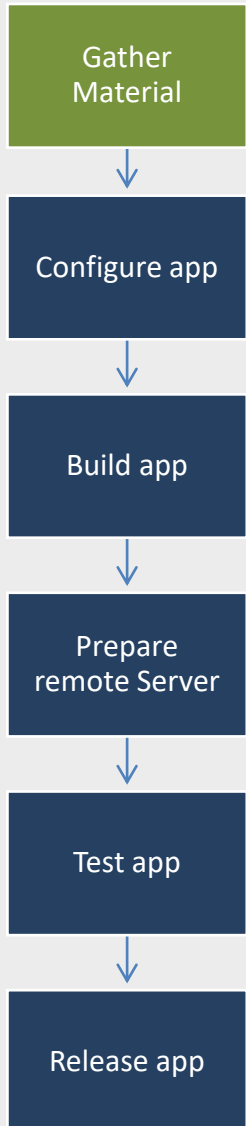
Repository

Mock Webservice → Unit-Tests

Webservice

Mock Webserver → Unit Tests

Publish your app



Application icon

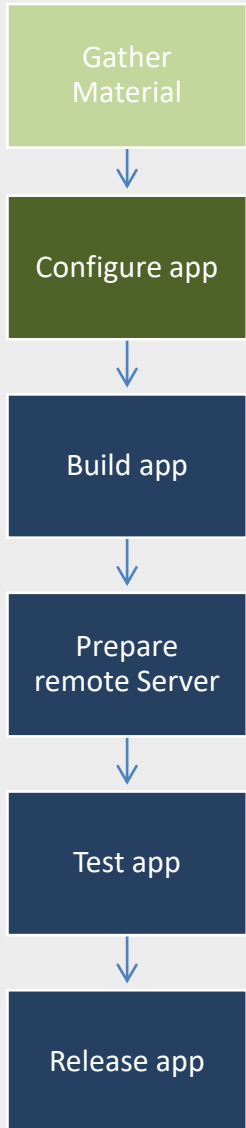
Users first interaction!

End user agreement (EULA)

Protect you against users

Miscellaneous material

Promotional material (screenshots, texts, ...)



Check package name

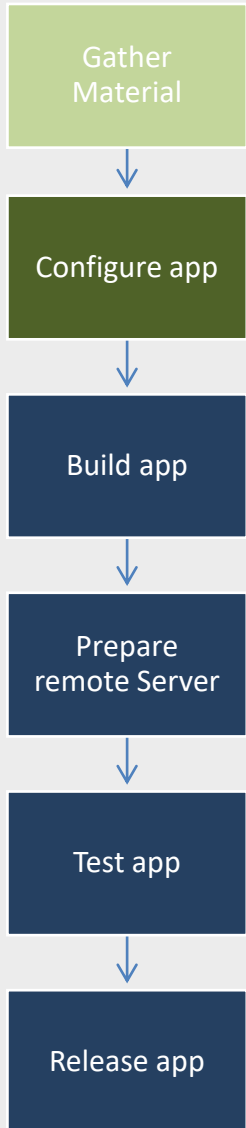
You cannot change it after publishing

Turn off logging

Remove all calls to the Log-class

Clean up project directories

Remove all unused files / libraries

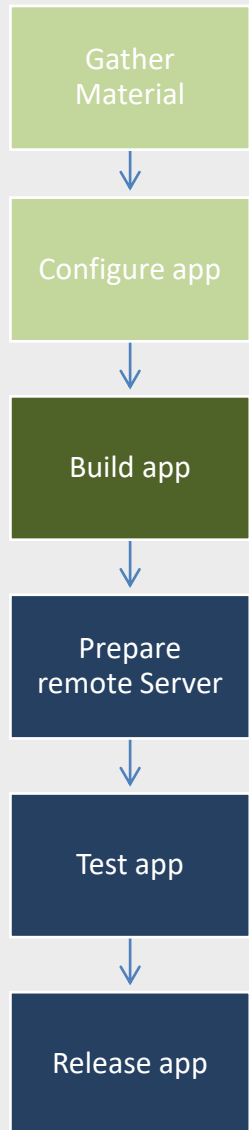


Review and update your manifest

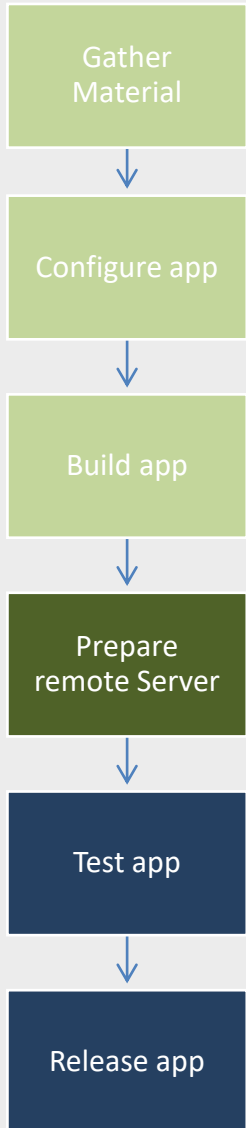
- Correct/review permissions
- Mandatory attributes (icon/ label)
- Recommended attributes (versionCode/ versionName)

Update URL's for server access

Publish your app



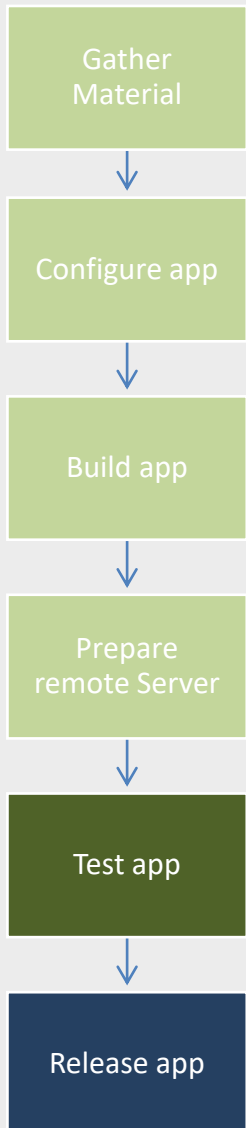
- Build the application to get the .apk-file
- Sign your application with your key
 - keytool / jarsigner



Check remote servers working

Check content

- No more test data
- Content uptodate

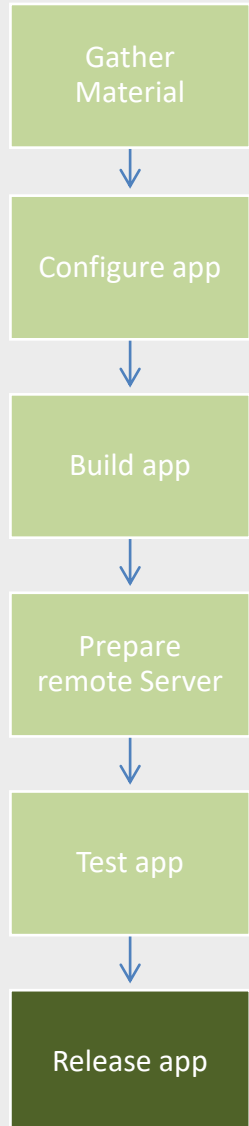


Test the ready apk file

- Install on devices
- Robotium may help

Test Clouds

- Run Testscripts on various devices
- Xamarin Test Cloud



Google Play Store

- Global / huge audience

App market places

- Amazon App store
- Samsung App store

E-Mail

- Specific receiver
- No protection from piracy

Website

- Provide apk as download
- User Opt-In required

Paixon

Google Play

Mehr als 3 Millionen apps

Mehr als 1 Milliarde aktive Benutzer

Requirements as publisher

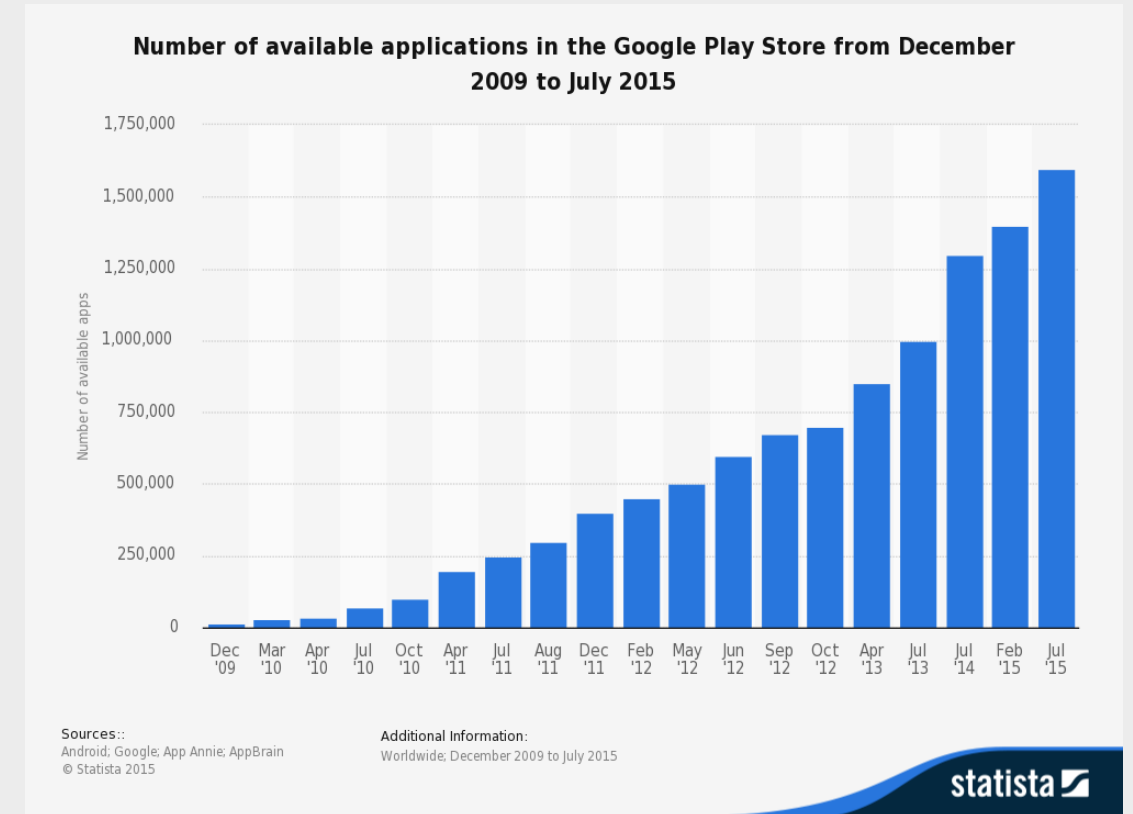
Google Play Publisher Account (25\$)

Google Wallet Merchant Account

Requirements as user

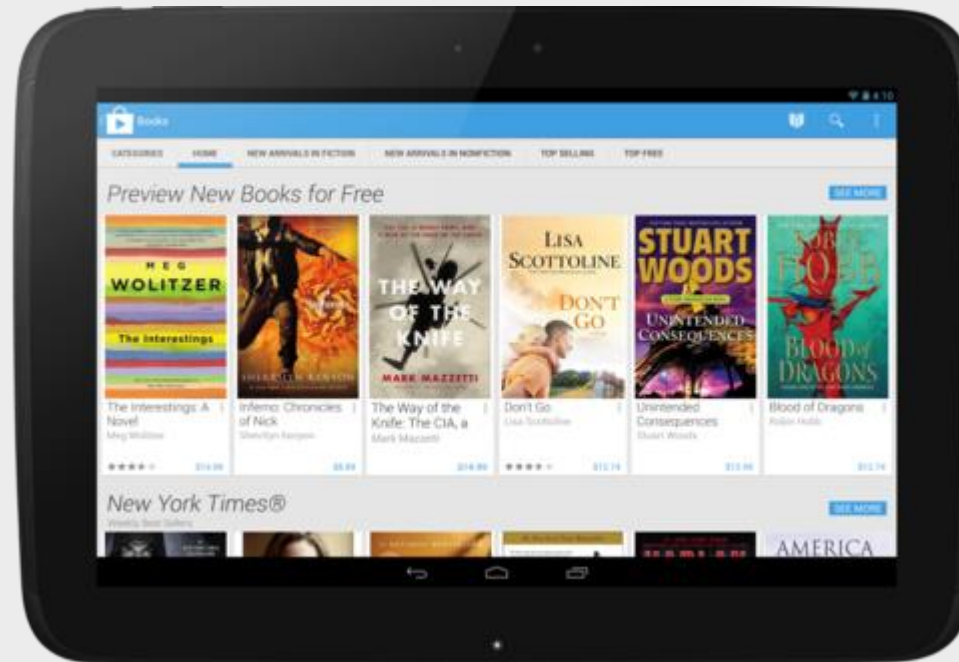
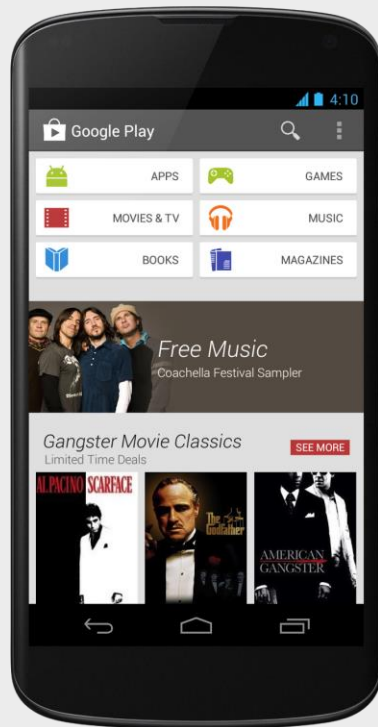
Google Account

Google Wallet Merchant Account



Google Play – App

Paixon



The screenshot shows a web browser window with the Google Play Store interface. The address bar displays the URL <https://play.google.com/store/apps/details?id=ch.sbb.mobile.android.b2c>. The page header includes the Google Play logo, a search bar with the text "Suchen", and user information for "Stefan". The left sidebar contains navigation links: "Meine Apps", "Einkaufen", "Spiele", "Familie", "Empfehlungen", "Meine Play-Aktivitäten", "Meine Wunschliste", "Code einlösen", "Geschenkkarte kaufen", and "Leitfaden für Eltern". The main content area displays the "SBB Mobile" app by "Schweizerische Bundesbahnen SBB" under the category "Verkehr". The app has a rating of 4.5 stars from 14,606 reviews and is marked as "PEGI 3". A green "Installiert" button indicates the app is already installed. Below the app title, there are three preview images: a man in a suit, a train, and a mobile app interface showing a "Fahrplan" (timetable) and "Übers" (overview) section. The right sidebar shows a "Ähnlich" (Similar) section with a "Mehr" button and a preview of the "SBB Mobile Business" app, which is marked as "KOSTENLOS" (Free).

SBB Mobile – Android-Ap

Stefan

Suchen

Google Play

Kategorien

Startseite

Apps-Top-Charts

Neuerscheinungen

Meine Apps

Einkaufen

Spiele

Familie

Empfehlungen

Meine Play-Aktivitäten

Meine Wunschliste

Code einlösen

Geschenkkarte kaufen

Leitfaden für Eltern

SBB Mobile

Schweizerische Bundesbahnen SBB

Verkehr

★★★★☆ 14.606

PEGI 3

Diese App ist mit allen deinen Geräten kompatibel.

Installiert

Ähnlich

Mehr

SBB Mobile Business

Schweizerische Bundesbahnen SBB

★★★★☆ KOSTENLOS

Fahrplan App

Technische Infos

API

<http://transport.opendata.ch/>

Ressourcen

/locations	Findet ÖV Stationen
/connections	Findet Verbindungen
/stationboard	Findet Verbindungen ab einer Station

Create APK File

Android Studio → Build → Generate Signed APK

Export Code

File → Export to ZIP File

Paixon

Finito!