

Pokémon App Dokumentation

C:\Users\Isa\AndroidStudioProjects\MyApplication

Ersteller: Isabell Noack

Projektstart: 18.10.2023

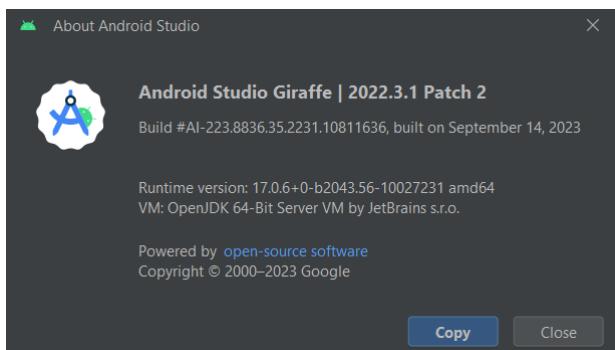
Modul: Mobile Apps Entwicklung

Prof. Dr.-Ing. Michael Stepping

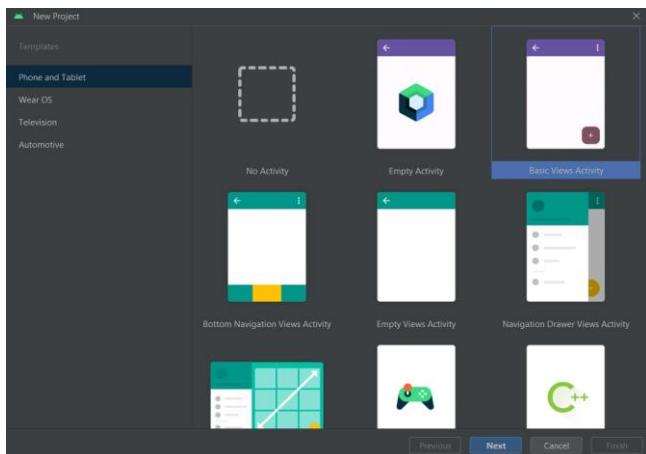
Ziel: Erstellung einer Mobile Android Applikation zum Aufrufen und detaillierten Einsehen von Pokémon

Projekt erstellen

Version: Android Studio Giraffe | 2022.3.1 Patch 2



Template benutzt (Basic Views Activity)



Package: com.isabellnoack.myapp

Github Repo erstellt

<https://github.com/IsaNck1/PokeApp>

PokéAPI

<https://pokeapi.co/>

Um Daten zu Pokémon zu bekommen, wird PokéAPI benutzt. Es ist kostenlos und open-source.

PokéAPI stellt hierfür JSON Daten im Internet bereit. Beispiel: <https://pokeapi.co/api/v2/pokemon/>

API JSON schön auslesen

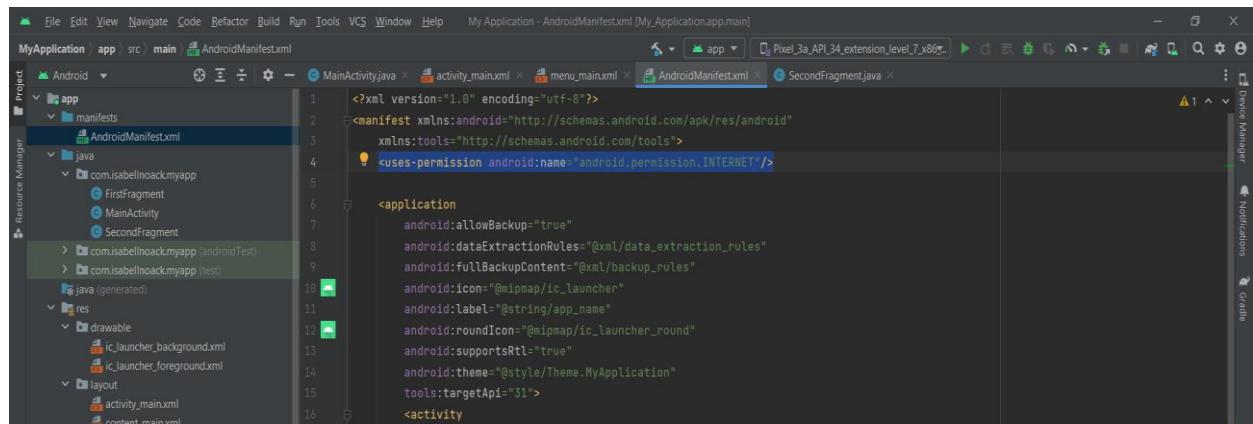
Hier reinkopieren, um es schön zu formatieren: <https://jsonformatter.curiousconcept.com/#>

Verwendung des Internets

AndroidManifest.xml

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

Berechtigung das Internet zu benutzen



PokeKotlin übernimmt das Parsen der JSON-Daten

PokeKotlin macht die JSON-Daten zu Java Instanzen

Für Java wird Reactor oder PokeKotlin benutzt, ich benutze PokeKotlin

Wrapper Libraries

- Node Server-side with auto caching: [Pokedex Promise v2](#) by Thomas Asadurian and Alessandro Pezzé
- Browser-side with auto caching: [pokeapi-js-wrapper](#) by Alessandro Pezzé
- Python 3 with auto caching: [PokeBase](#) by Greg Hilmes
- Python 2/3 with auto caching: [PokePy](#) by Paul Hallett
- [Kotlin \(and Java\)](#): [PokeKotlin](#) by sargunster
- Java (Spring Boot) with auto caching: [pokeapi-reactor](#) by Benjamin Churchill
- .NET (C#, VB, etc): [PokeApi.NET](#) by PoroCYon
- .NET Standard: [PokeApiNet](#) by mtrdp642
- Swift: [PokemonAPI](#) by kinkofer
- PHP: [PokePHP](#) by Dan Rovito
- PHP: [PHPKéAPI](#) by Imerotta
- Ruby: [Poke-API-V2](#) by rdavid1099
- Go: [pokeapi-go](#) by mtslzs
- Crystal: [pokeapi](#) by henrikac
- Typescript with auto caching: [Pokenode-ts](#) by Gabb-c
- Rust with auto caching: [Rustemon](#) by mlemesle
- Asynchronous Python wrapper with auto caching: [aiopokeapi](#) by beastmatser
- Scala 3 with auto caching: [pokeapi-scala](#) by Juliano Alves
- Elixir wrapper with auto caching: [Max-Elixir-PokeAPI](#) by Henrique Artur

<https://pokeapi.co/docs/v2>

<https://github.com/PokeAPI/pokekotlin>

Von Github die Bibliothek (Dependencies) und Repos in unser Projekt rein packen

<https://github.com/PokeAPI/pokekotlin/releases>

```
repositories {
```

```
    mavenCentral()
```

```

        url = uri("https://jitpack.io")
    }

dependencies {
    implementation 'com.github.PokeAPI:pokekotlin:2.3.1'
}

```

2.3.1

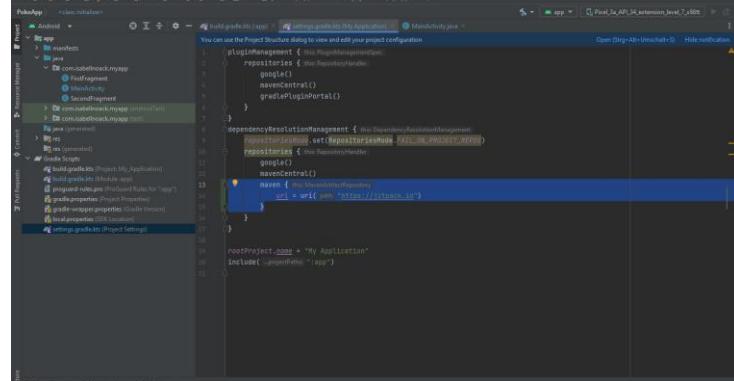
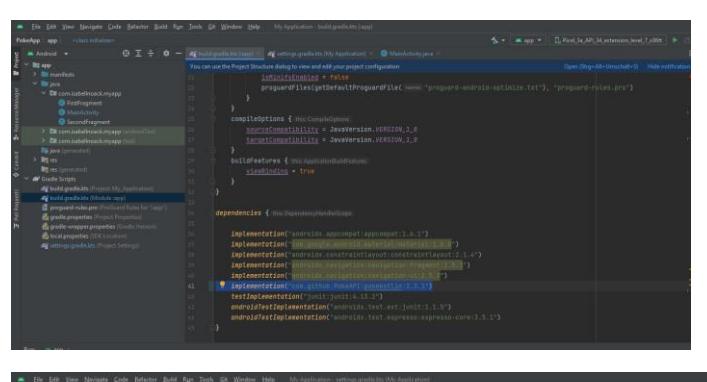
Bintray proved to be unreliable, so the package is now published on GitHub packages. It's also updated to the latest dependencies (kotlin and retrofit) and updated to pass tests on the latest pokeapi data.

EDIT: Turns out GH Packages requires auth even for public packages, so use JitPack:

```

repositories {
    mavenCentral()
    maven { url 'https://jitpack.io' }
}
dependencies {
    implementation 'com.github.PokeAPI:pokekotlin:<VERSION>'
}

```



Mit Alt + Enter Klassen importieren

```

import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

import com.isabellnoack.myapp.databinding.ActivityMainBinding;

import android.view.Menu;
import android.view.MenuItem;

import me.sargunvohra.lib.pokekotlin.client.PokeApi;
import me.sargunvohra.lib.pokekotlin.client.PokeApiClient;
import me.sargunvohra.lib.pokekotlin.model.PokemonSpecies;

3 usages  ↳ Isa *
public class MainActivity extends AppCompatActivity {

    3 usages
    private AppBarConfiguration appBarConfiguration;
    4 usages
    private ActivityMainBinding binding;

    ↳ Isa *
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

Code ausprobieren

```

3 usages  ↳ Isa *
public class MainActivity extends AppCompatActivity {

    3 usages
    private AppBarConfiguration appBarConfiguration;
    4 usages
    private ActivityMainBinding binding;

    ↳ Isa *
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        PokeApi pokeApi = new PokeApiClient();
        PokemonSpecies bulbasaur = pokeApi.getPokemonSpecies(1);
        System.out.println(bulbasaur);

        binding = ActivityMainBinding.inflate(LayoutInflaterCompat.createFrom(this));
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment_main);
        appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();
        NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);
    }
}

```

Threads

Fehlermeldung und App Crash:

FATAL EXCEPTION: main ... Caused by: android.os.NetworkOnMainThreadException

Grund: Internet Kommunikation auf UI Thread ist böse.

Lösung: Abfrage auf neuen Thread

Teil 2:

- Erweiterung der Fussball-App.
- Datenbestände verschieben in eine zu erstellende Applikationsklasse.
- Abholung der Spielstände aus dem Datenbestand der Openliga-DB.
- Hierzu einbau eines Thread für die Netzwerkkommunikation
- Parsen der JSON Daten und überführen in die eigenen Daten.
- Abfrage des Spieltages und entsprechende Darstellung der Daten in den Dialogen
- Einbau eines GPS-Sensors und Darstellung der Position (als Koordinaten) oder auch als Kartenausschnitt (Open Street Map oder Google Maps)
- Abgabe bis Vorlesungsende

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help My Application - MainActivity.java [My_Application.app.main]
PokeApp app src main java com.isabellnoack.myapp MainActivity onCreate
Android build.gradle.kts (app) settings.gradle.kts (My Application) MainActivity.java
Project Resource Manager
  app
    manifests
    java
      com.isabellnoack.myapp
        FirstFragment
        MainActivity
        SecondFragment
    com.isabellnoack.myapp (androidTest)
    com.isabellnoack.myapp (test)
    java (generated)
    res
    res (generated)
Gradle Scripts
  build.gradle.kts (Project: My Application)
  build.gradle.kts (Module: app)
  proguard-rules.pro (ProGuard Rules for ":app")
  gradle.properties (Project Properties)
  gradle-wrapper.properties (Gradle Version)
  local.properties (SDK Location)
  settings.gradle.kts (Project Settings)

private AppBarConfiguration appBarConfiguration;
private ActivityMainBinding binding;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    new Thread() -> {
        PokeApi pokeApi = new PokeApiClient();
        PokemonSpecies bulbasaur = pokeApi.getPokemonSpecies(1);
        System.out.println(bulbasaur);
    }.start();

    binding = ActivityMainBinding.inflate(LayoutInflaterCompat.createFrom(this));
    setContentView(binding.getRoot());

    setSupportActionBar(binding.toolbar);

    NavController navController = Navigation.findNavController(activity: this, R.id.nav_host_fragment_content_main);
    appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();
    NavigationUI.setupActionBarWithNavController(activity: this, navController, appBarConfiguration);
}

@Isa
```

Test Bulbasaur als Text erstellen und ersetzen

Dafür zuerst das Bulbasaur Test in FirstFragment gepackt, dort dann mit

```
binding.textViewFirst.setText(bulbasaur.toString()))
```

ersetzt.

The screenshot shows the Android Studio interface with the code editor open to `FirstFragment.java`. The code implements a `Fragment` with a button that navigates to a second fragment. The Java code is annotated with Lombok annotations (`@Data`, `@RequiredArgsConstructor`, `@NoArgsConstructor`, `@AllArgsConstructor`). The `binding` variable is generated by Data Binding. The `onViewCreated` method sets up an `onClick` listener for the button. The `onDestroyView` method releases the binding object.

```
    binding = FragmentFirstBinding.inflate(inflater, container, attachToParent: false);
    return binding.getRoot();
}

@Override
public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

    binding.buttonFirst.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            NavHostFragment.findNavController(FragmentFirstFragment.this)
                .navigate(R.id.action_FirstFragment_to_SecondFragment);
        }
    });

    new Thread() -> {
        PokeApiClient pokeApiClient = new PokeApiClient(); // Instanz einer Klasse erstellen
        PokemonSpecies bulbasaur = pokeApiClient.getPokemonSpecies(1); // Funktion der neuen Instanz PokeAPI aufrufen
        Ability ability = pokeApiClient.getAbility(1);
        System.out.println(ability);
        getMainActivity().runOnUiThread() -> {
            // Hier UI verändern: show text on UI/button
            binding.textViewFirst.setText(bulbasaur.toString());
        };
    }.start();
}

@Override
public void onDestroyView() {
    super.onDestroyView();
    binding = null;
}
```

The right side of the screen displays a preview of the application's first fragment. It shows a single button labeled "BULBASAUR". A tooltip for the button provides detailed information about the `Bulbasaur` species, including its name, language, version, and category. The preview is set against a background of a sun rising over a field of grass.

```

new Thread(() -> {
    PokeApi pokeApi = new PokeApiClient(); //Instanz einer Klasse erstellen
    PokemonSpecies bulbasaur = pokeApi.getPokemonSpecies( 1); //Funktion der
                                                               // neuen Instanz pokeAPI aufrufen
    Ability a = pokeApi.getAbility( 1);
    System.out.println(a);
    getActivity().runOnUiThread(() -> {
        // Hier UI verändern: show text on UI/button
        binding.textViewFirst.setText(bulbasaur.toString());
    });
}).start();

```

Sprach-Einstellungen

Sprach-Spezifisch unter res => values => strings.xml

In strings.xml(de) Steht dann der Text für alles mit Deutschem System, ansonsten wird die Allgemeine strings.xml benutzt

```

<string name="hello_world">Hello World</string>      strings.xml
<string name="hello_world">Hallo Welt</string>      strings.xml(de)

```

JSON auslesen (JSONReader)

Dokumentation zum JSON Reader <https://developer.android.com/reference/android/util/JsonReader>

Standard mäßig in Android drinnen. Klasse importieren da es in einem anderen Package ist.

```
import android.util.JsonReader;
```

JSON Reader in der Klasse PokeAPI (PokeAPI.java file)

JSONObject vs. JSONReader

Proff nimmt JSONObject

geht aber beides

JSONObject liest JSON auf einmal aus

JSONReader geht Key für Key durch

=> bei vielen Daten hat der JSONReader einen geringeren Speicherbedarf, also kann bessere Effizienz und Leistung geben

=> mir wurde auch empfohlen von mehreren Leuten den zu nutzen, weil die damit mehr erfahrung hatten

Funktion requestPokemon (Ablauf JSONReader)

Funktion requestPokemon wird aufgerufen. (STRG + Links Klick zum Finden, wo es aufgerufen wird)

```
requestPokemon
```

Name der Funktion

```
(int id)
```

Parameter

```
Pokemon
```

Rückgabewert (Eine Instanz der Klasse Pokemon)

```
throws IOException
```

Ausnahme (Verbindung bricht ab; übers Ende hinauslesen; Datei nicht gefunden)

```
//1
public Pokemon requestPokemon(int id) throws IOException {
    //Wo lesen wir unsere Daten her
    String url = "https://pokeapi.co/api/v2/pokemon/" + id; //Datenquelle
    JsonReader reader;
    reader = requestJsonReader(url);
    //Interpretiert Text aus Internet als JSON
    return readPokemon(reader); //Funktion für JSON in Pokemon Instanz
}
```

1. Festlegung, woher die Daten kommen: URL
2. JSON Reader nimmt die URL und liest von dort die Daten aus
3. Neue Instanz der Klasse JSON Reader „reader“ (Variable reader ist noch undefined)
4. Wir Speichern den Rückgabe wert der Funktion requestJsonReader(url) in den reader
5. **Funktion requestJsonReader(url) gibt uns den bufferedReader in der Variable reader zurück**
6. Wir rufen die Funktion readPokemon(JSONReader reader) mit dem Wert reader auf
7. **Funktion Pokemon readPokemon(JsonReader reader)**
8. Der Rückgabewert wird zurückgegeben

Funktion JsonReader requestJsonReader(String url)

```
//2
JsonReader requestJsonReader(String url) throws IOException { //JSON lesen
    HttpURLConnection connection = (HttpURLConnection) new
    URL(url).openConnection();
    connection.setRequestProperty("User-Agent", "Mozilla/5.0 (Windows NT
    10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0
    Safari/537.36");
    BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(
    connection.getInputStream(), "UTF-8")); //Datenstrom als UTF-8 lesen
    return new JsonReader(bufferedReader); //Text/JSON als Antwort
}
```

```
JsonReader requestJsonReader(String url) throws IOException { //JSON lesen
Funktion mit Eingabewert String url
```

```
HttpURLConnection connection = (HttpURLConnection) new
URL(url).openConnection();
```

Wir haben die Variable connection der Klasse HttpURLConnection.

Wir machen aus dem String url eine URL(url) damit wir .openConnection() damit aufrufen können.

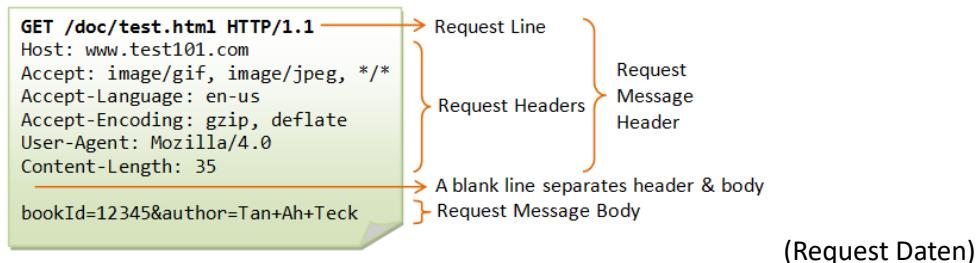
(.openConnection() ist in der Klasse URL definiert)

(Sie String url kommt von: String url = "https://pokeapi.co/api/v2/pokemon/" + id;)

```
(HttpURLConnection) new URL(url).openConnection()
```

Die open Connection kann viele verschiedene Typen von urls benutzen. Wir wissen, dass wir eine HttpURLConnection haben. Das nennt man casten (Datentyp einer Oberklasse zu spezieller Unterklasse, damit man das als Unterklasse benutzen kann muss man den Datentyp der variable ändern)

HttpURLConnection unterstützt Header (weil das http Protokoll Header unterstützt (Zusatzdaten))



```
connection.setRequestProperty("User-Agent", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36");
```

Funktion setRequestProperty der Klasse HttpURLConnection

Aussage: Ich unterstütze alle Features von Firefox der Version 5, Ich unterstütze alle Features die die Version „Chrome 118“, „Safari 537“ etc. hat.

Das Bild zeigt den Android Developers Reference-Webscraper. Die linke Seite zeigt die Klassenhierarchie von JsonReader bis hinunter zu den Implementierungen von HttpURLConnection. Die rechte Seite zeigt die Dokumentation für die Klasse JsonReader, einschließlich Methodenbeschreibungen wie `parse` und `parseString`.

Von hier genommen. Das machen alle Browser so.

Ansonsten antwortet die API nicht.

```
BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(connection.getInputStream(), "UTF-8"));
```

Variable der Klasse BufferedReader „bufferedReader“ deklariert

```
new BufferedReader(new InputStreamReader(connection.getInputStream(), "UTF-8"));
```

Funktion connection.getInputStream() wird aufgerufen

Die Verbindung wird hergestellt und die Request Daten werden zum Server geschickt.

Wir bekommen einen InputStream zurück. (von der Funktion .getInputStream()) (Datenstrom als UTF-8 Format)

Der InputStream wird zu einem BufferedReader konvertiert. (und in unserem bufferedReader gespeichert)

```
return new JsonReader(bufferedReader);
```

Wir bekommen den bufferedReader als Rückgabewert.

Funktion Pokemon readPokemon(JsonReader reader)

```
//3
Pokemon readPokemon(JsonReader reader) throws IOException {
    Pokemon pokemon = new Pokemon();
    reader.beginObject();
    while (reader.hasNext()) {
        switch (reader.nextName()) {
            case "name":
                pokemon.name = reader.nextString();
                break;
            case "base_experience":
                pokemon.baseExperience = reader.nextInt();
                break;
            default:
                reader.skipValue();
        }
    }
    reader.endObject();
    return pokemon;
}
```

```
Pokemon readPokemon(JsonReader reader) throws IOException
```

Funktion readPokemon mit Input JsonReader reader

Rückgabetyp: Pokemon

```
Pokemon pokemon = new Pokemon();
```

Pokemon erstellen (Instanz der Klasse Pokemon mit der Bezeichnung „pokemon“)

Pokemon = Typ der Variable

pokemon = Name der Variable

new Pokemon() = Konstruktor (Besondere Funktion)

```
reader.beginObject();
```

Funktion beginObject() wird aufgerufen für den reader

Es liest die Daten vom Server bis er „{“ (Object) findet

```
while (reader.hasNext()) {
    switch (reader.nextName()) {
        case "name":
            pokemon.name = reader.nextString(); //Formatierung: Strg+Alt+O
```

```

        break;
    case "base_experience":
        pokémon.baseExperience = reader.nextInt();
        break;
    default:
        reader.skipValue();
    }
}

```

hasNext() schaut ob Klammer und Kommas (Symbole heißen Tokens) da sind um das Objekt/Eigenschaften zu lesen => sagt also ob es nächste Eigenschaft/Element gibt

reader.nextName() In dem Objekt sagt uns die Funktion wie die Eigenschaft heißt

Wenn er „name“ liest, wird in pokémon.name der darauf folgende String gespeichert

Wenn er „base_experience“ liest, wird in pokémon.baseExperience der darauf folgende Integer gespeichert

Ansonsten werden die Werte übersprungen.

```
reader.endObject();
```

Objekt zuende gelesen

```
return pokémon;
```

Die Instanz der Klasse Pokemon mit den zugewiesenen Werten wird ausgegeben.

Klassen

Klasse Pokemon

```

public class Pokemon {
    public String name;
    public int baseExperience;

    //magic mit: tostring, drei runter und enter (generate via wizard)
    @Override
    public String toString() {
        return "Pokemon{" +
            "name='" + name + '\'' +
            ", baseExperience=" + baseExperience +
            '}';
    }
}

```

Variable String name

Variable Integer baseExperience

```
public String toString()
```

Der Wizard erstellt uns die `toString()` Funktion. Die ist da um Dinge in der Konsole schön ausgeben zu können, für die App ist es irrelevant.

Klasse Berry (Beispiel für Array)

```

public class Berry {
    public String name = "Unknown";
    public int size;
    public int smoothness;
    public ArrayList<BerryFlavor> flavors = new ArrayList<>();

    @Override
    public String toString() {
        return "Berry{" +

```

```

        "name=''" + name + '\'' +
        ", size=" + size +
        ", smoothness=" + smoothness +
        ", flavors=" + flavors +
        '}';
    }
}

```

In der Klasse Berry ist eine Liste von Objekten der Klasse BerryFlavor

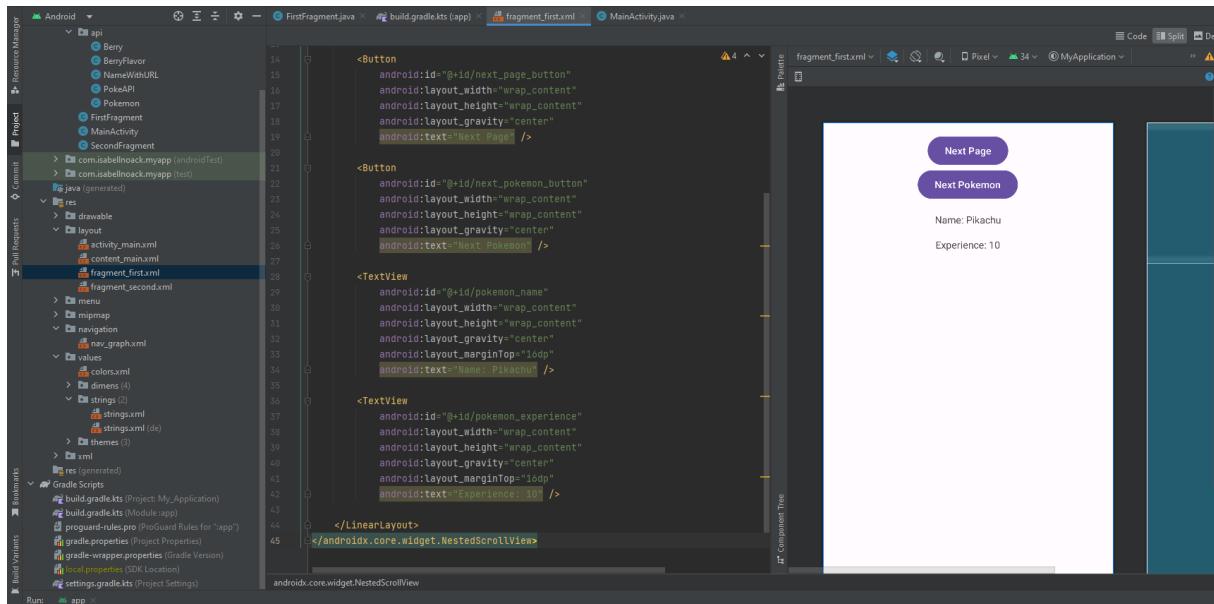
Jede Berry Instanz hat eine eigene Liste (array)

Notiz: Mit „static“ würde es für alle Berry eine gemeinsame Liste geben. Static heißt es braucht keine Instanz

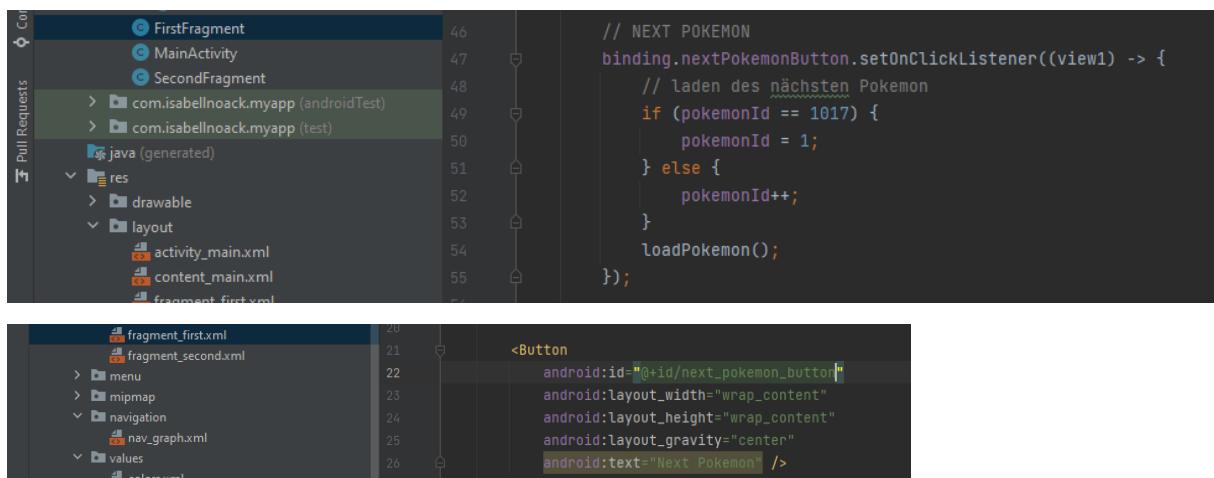
Feature: Next Pokemon Button

Button angelegt für nächstes Pokemon

Button im Layout angelegt, ID zum Verknüpfen



Click Listener setzen (Funktion, die ausgeführt wird, wenn Button gedrückt wird)



Wir rufen Funktion loadPokemon() auf mit dem binding.nextPokemonButton

Previous Button hinzugefügt auf die gleiche Art

Wrap Around (Bei last Pokemon (1017) -> next Pokemon ist 1)

Problem: App beim Knopfdruck (Previous Pokemon) abgestürzt, wenn man beim ersten Pokemon war. Ansonsten aht alles geklappt

Grund: Manche Pokemon haben bei der variablen Base_Experience den Wert null

The screenshot shows the Android Studio interface with the project structure on the left and Java code on the right. The project structure includes files like FirstFragment, MainActivity, SecondFragment, com.isabellnoack.myapp (androidTest and test), java (generated), res (drawable, layout, menu, mipmap, navigation, values, xml), and Gradle Scripts. The Java code handles the next and previous Pokemon buttons:

```
// NEXT POKEMON
binding.nextPokemonButton.setOnClickListener((view1) -> {
    // laden des nächsten Pokemon
    if (pokemonId == 1017) {
        pokemonId = 1;
    } else {
        pokemonId++;
    }
    loadPokemon();
});

//PREVIOUS POKEMON
binding.previousPokemonButton.setOnClickListener((view1) -> {
    // laden des vorherigen Pokemon
    if (pokemonId == 1) {
        pokemonId = 1017;
    } else {
        pokemonId--;
    }
    loadPokemon();
});
```

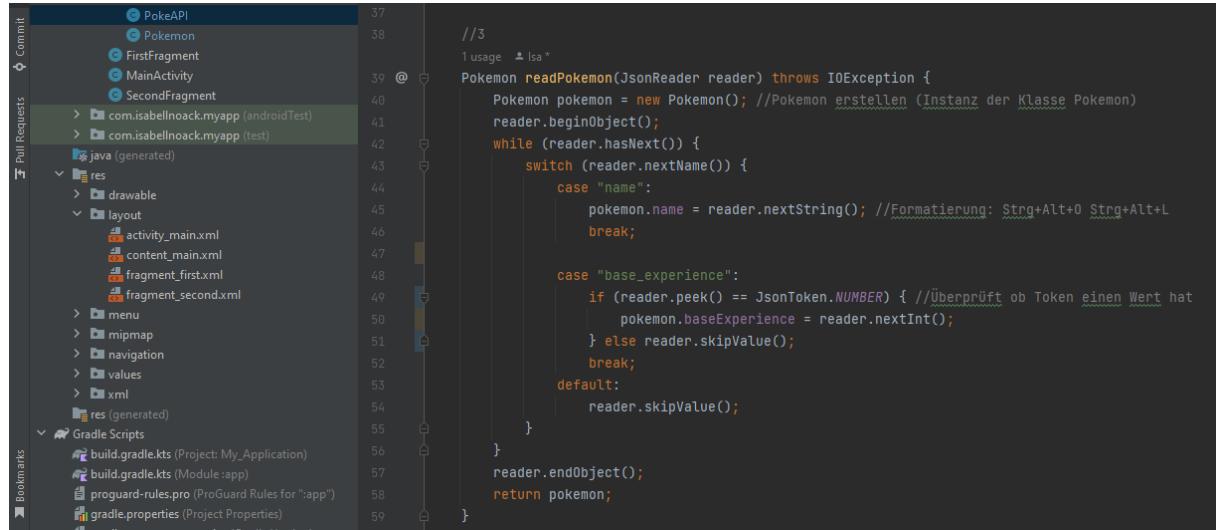
Wrap Around mit if und else:

```
// NEXT POKEMON
binding.nextPokemonButton.setOnClickListener((view1) -> {
    // laden des nächsten Pokemon
    if (pokemonId == 1017) {
        pokemonId = 1;
    } else {
        pokemonId++;
    }
    loadPokemon();
});

//PREVIOUS POKEMON
binding.previousPokemonButton.setOnClickListener((view1) -> {
    // laden des vorherigen Pokemon
    if (pokemonId == 1) {
        pokemonId = 1017;
    } else {
        pokemonId--;
    }
    loadPokemon();
});
```

Lösung:

Bei `readPokemon()` wenn `base_experience` ausgelesen wird, wird zuerst überprüft, ob der Token eine Zahl ist. Wenn nicht, wird der Wert übersprungen. (Hier wird dann der Standard Wert genommen, welcher bei der Deklaration der Variablen angegeben ist.)



```

37
38 //3
39 @ ... 1 usage  ↳ Is a *
40
41 Pokemon readPokemon(JsonReader reader) throws IOException {
42     Pokemon pokemon = new Pokemon(); //Pokemon erstellen (Instanz der Klasse Pokemon)
43     reader.beginObject();
44     while (reader.hasNext()) {
45         switch (reader.nextName()) {
46             case "name":
47                 pokemon.name = reader.nextString(); //Formatierung: Strg+Alt+0 Strg+Alt+L
48                 break;
49
50             case "base_experience":
51                 if (reader.peek() == JsonToken.NUMBER) { //Überprüft ob Token einen Wert hat
52                     pokemon.baseExperience = reader.nextInt();
53                 } else reader.skipValue();
54                 break;
55             default:
56                 reader.skipValue();
57         }
58     }
59     reader.endObject();
60     return pokemon;
}

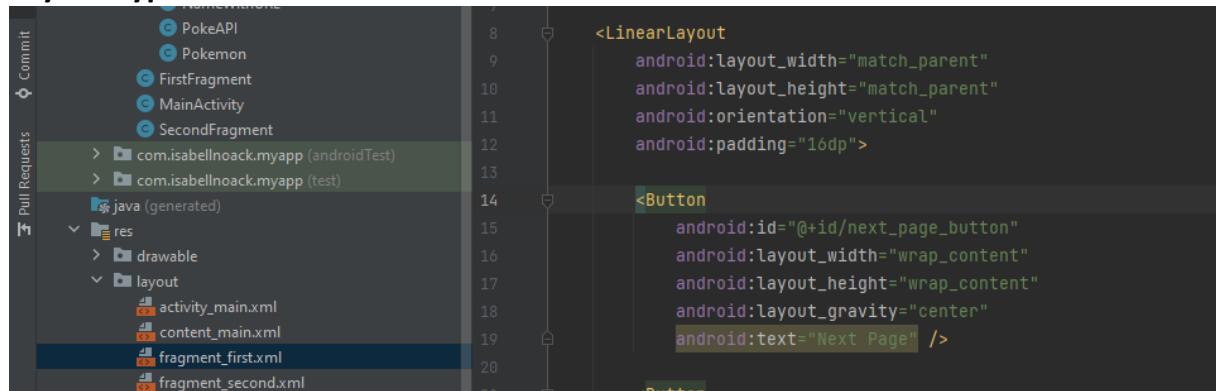
```

```

case "base_experience":
    if (reader.peek() == JsonToken.NUMBER) { //Überprüft ob Token einen
Wert hat
        pokemon.baseExperience = reader.nextInt();
    } else reader.skipValue();
    break;

```

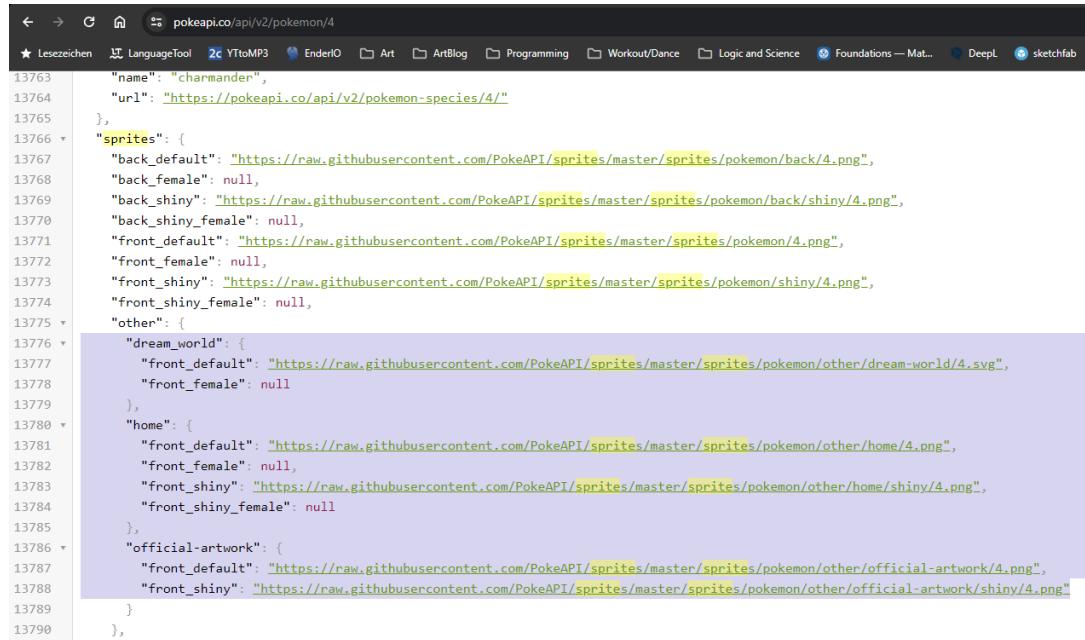
Layout Type



```

8
9 <LinearLayout
10    android:layout_width="match_parent"
11    android:layout_height="match_parent"
12    android:orientation="vertical"
13    android:padding="16dp">
14
15    <Button
16        android:id="@+id/next_page_button"
17        android:layout_width="wrap_content"
18        android:layout_height="wrap_content"
19        android:layout_gravity="center"
20        android:text="Next Page" />
21
22    <Text ...
23
24    <Text ...
25
26    <Text ...
27
28    <Text ...
29
30    <Text ...
31
32    <Text ...
33
34    <Text ...
35
36    <Text ...
37
38    <Text ...
39
40    <Text ...
41
42    <Text ...
43
44    <Text ...
45
46    <Text ...
47
48    <Text ...
49
50    <Text ...
51
52    <Text ...
53
54    <Text ...
55
56    <Text ...
57
58    <Text ...
59
60    <Text ...
61
62    <Text ...
63
64    <Text ...
65
66    <Text ...
67
68    <Text ...
69
70    <Text ...
71
72    <Text ...
73
74    <Text ...
75
76    <Text ...
77
78    <Text ...
79
80    <Text ...
81
82    <Text ...
83
84    <Text ...
85
86    <Text ...
87
88    <Text ...
89
90    <Text ...
91
92    <Text ...
93
94    <Text ...
95
96    <Text ...
97
98    <Text ...
99
100   <Text ...
101
102   <Text ...
103
104   <Text ...
105
106   <Text ...
107
108   <Text ...
109
110   <Text ...
111
112   <Text ...
113
114   <Text ...
115
116   <Text ...
117
118   <Text ...
119
120   <Text ...
121
122   <Text ...
123
124   <Text ...
125
126   <Text ...
127
128   <Text ...
129
130   <Text ...
131
132   <Text ...
133
134   <Text ...
135
136   <Text ...
137
138   <Text ...
139
140   <Text ...
141
142   <Text ...
143
144   <Text ...
145
146   <Text ...
147
148   <Text ...
149
150   <Text ...
151
152   <Text ...
153
154   <Text ...
155
156   <Text ...
157
158   <Text ...
159
160   <Text ...
161
162   <Text ...
163
164   <Text ...
165
166   <Text ...
167
168   <Text ...
169
170   <Text ...
171
172   <Text ...
173
174   <Text ...
175
176   <Text ...
177
178   <Text ...
179
180   <Text ...
181
182   <Text ...
183
184   <Text ...
185
186   <Text ...
187
188   <Text ...
189
190   <Text ...
191
192   <Text ...
193
194   <Text ...
195
196   <Text ...
197
198   <Text ...
199
200   <Text ...
201
202   <Text ...
203
204   <Text ...
205
206   <Text ...
207
208   <Text ...
209
210   <Text ...
211
212   <Text ...
213
214   <Text ...
215
216   <Text ...
217
218   <Text ...
219
220   <Text ...
221
222   <Text ...
223
224   <Text ...
225
226   <Text ...
227
228   <Text ...
229
230   <Text ...
231
232   <Text ...
233
234   <Text ...
235
236   <Text ...
237
238   <Text ...
239
240   <Text ...
241
242   <Text ...
243
244   <Text ...
245
246   <Text ...
247
248   <Text ...
249
250   <Text ...
251
252   <Text ...
253
254   <Text ...
255
256   <Text ...
257
258   <Text ...
259
260   <Text ...
261
262   <Text ...
263
264   <Text ...
265
266   <Text ...
267
268   <Text ...
269
270   <Text ...
271
272   <Text ...
273
274   <Text ...
275
276   <Text ...
277
278   <Text ...
279
280   <Text ...
281
282   <Text ...
283
284   <Text ...
285
286   <Text ...
287
288   <Text ...
289
290   <Text ...
291
292   <Text ...
293
294   <Text ...
295
296   <Text ...
297
298   <Text ...
299
299   <Text ...
300
300   <Text ...
301
301   <Text ...
302
302   <Text ...
303
303   <Text ...
304
304   <Text ...
305
305   <Text ...
306
306   <Text ...
307
307   <Text ...
308
308   <Text ...
309
309   <Text ...
310
310   <Text ...
311
311   <Text ...
312
312   <Text ...
313
313   <Text ...
314
314   <Text ...
315
315   <Text ...
316
316   <Text ...
317
317   <Text ...
318
318   <Text ...
319
319   <Text ...
320
320   <Text ...
321
321   <Text ...
322
322   <Text ...
323
323   <Text ...
324
324   <Text ...
325
325   <Text ...
326
326   <Text ...
327
327   <Text ...
328
328   <Text ...
329
329   <Text ...
330
330   <Text ...
331
331   <Text ...
332
332   <Text ...
333
333   <Text ...
334
334   <Text ...
335
335   <Text ...
336
336   <Text ...
337
337   <Text ...
338
338   <Text ...
339
339   <Text ...
340
340   <Text ...
341
341   <Text ...
342
342   <Text ...
343
343   <Text ...
344
344   <Text ...
345
345   <Text ...
346
346   <Text ...
347
347   <Text ...
348
348   <Text ...
349
349   <Text ...
350
350   <Text ...
351
351   <Text ...
352
352   <Text ...
353
353   <Text ...
354
354   <Text ...
355
355   <Text ...
356
356   <Text ...
357
357   <Text ...
358
358   <Text ...
359
359   <Text ...
360
360   <Text ...
361
361   <Text ...
362
362   <Text ...
363
363   <Text ...
364
364   <Text ...
365
365   <Text ...
366
366   <Text ...
367
367   <Text ...
368
368   <Text ...
369
369   <Text ...
370
370   <Text ...
371
371   <Text ...
372
372   <Text ...
373
373   <Text ...
374
374   <Text ...
375
375   <Text ...
376
376   <Text ...
377
377   <Text ...
378
378   <Text ...
379
379   <Text ...
380
380   <Text ...
381
381   <Text ...
382
382   <Text ...
383
383   <Text ...
384
384   <Text ...
385
385   <Text ...
386
386   <Text ...
387
387   <Text ...
388
388   <Text ...
389
389   <Text ...
390
390   <Text ...
391
391   <Text ...
392
392   <Text ...
393
393   <Text ...
394
394   <Text ...
395
395   <Text ...
396
396   <Text ...
397
397   <Text ...
398
398   <Text ...
399
399   <Text ...
400
400   <Text ...
401
401   <Text ...
402
402   <Text ...
403
403   <Text ...
404
404   <Text ...
405
405   <Text ...
406
406   <Text ...
407
407   <Text ...
408
408   <Text ...
409
409   <Text ...
410
410   <Text ...
411
411   <Text ...
412
412   <Text ...
413
413   <Text ...
414
414   <Text ...
415
415   <Text ...
416
416   <Text ...
417
417   <Text ...
418
418   <Text ...
419
419   <Text ...
420
420   <Text ...
421
421   <Text ...
422
422   <Text ...
423
423   <Text ...
424
424   <Text ...
425
425   <Text ...
426
426   <Text ...
427
427   <Text ...
428
428   <Text ...
429
429   <Text ...
430
430   <Text ...
431
431   <Text ...
432
432   <Text ...
433
433   <Text ...
434
434   <Text ...
435
435   <Text ...
436
436   <Text ...
437
437   <Text ...
438
438   <Text ...
439
439   <Text ...
440
440   <Text ...
441
441   <Text ...
442
442   <Text ...
443
443   <Text ...
444
444   <Text ...
445
445   <Text ...
446
446   <Text ...
447
447   <Text ...
448
448   <Text ...
449
449   <Text ...
450
450   <Text ...
451
451   <Text ...
452
452   <Text ...
453
453   <Text ...
454
454   <Text ...
455
455   <Text ...
456
456   <Text ...
457
457   <Text ...
458
458   <Text ...
459
459   <Text ...
460
460   <Text ...
461
461   <Text ...
462
462   <Text ...
463
463   <Text ...
464
464   <Text ...
465
465   <Text ...
466
466   <Text ...
467
467   <Text ...
468
468   <Text ...
469
469   <Text ...
470
470   <Text ...
471
471   <Text ...
472
472   <Text ...
473
473   <Text ...
474
474   <Text ...
475
475   <Text ...
476
476   <Text ...
477
477   <Text ...
478
478   <Text ...
479
479   <Text ...
480
480   <Text ...
481
481   <Text ...
482
482   <Text ...
483
483   <Text ...
484
484   <Text ...
485
485   <Text ...
486
486   <Text ...
487
487   <Text ...
488
488   <Text ...
489
489   <Text ...
490
490   <Text ...
491
491   <Text ...
492
492   <Text ...
493
493   <Text ...
494
494   <Text ...
495
495   <Text ...
496
496   <Text ...
497
497   <Text ...
498
498   <Text ...
499
499   <Text ...
500
500   <Text ...
501
501   <Text ...
502
502   <Text ...
503
503   <Text ...
504
504   <Text ...
505
505   <Text ...
506
506   <Text ...
507
507   <Text ...
508
508   <Text ...
509
509   <Text ...
510
510   <Text ...
511
511   <Text ...
512
512   <Text ...
513
513   <Text ...
514
514   <Text ...
515
515   <Text ...
516
516   <Text ...
517
517   <Text ...
518
518   <Text ...
519
519   <Text ...
520
520   <Text ...
521
521   <Text ...
522
522   <Text ...
523
523   <Text ...
524
524   <Text ...
525
525   <Text ...
526
526   <Text ...
527
527   <Text ...
528
528   <Text ...
529
529   <Text ...
530
530   <Text ...
531
531   <Text ...
532
532   <Text ...
533
533   <Text ...
534
534   <Text ...
535
535   <Text ...
536
536   <Text ...
537
537   <Text ...
538
538   <Text ...
539
539   <Text ...
540
540   <Text ...
541
541   <Text ...
542
542   <Text ...
543
543   <Text ...
544
544   <Text ...
545
545   <Text ...
546
546   <Text ...
547
547   <Text ...
548
548   <Text ...
549
549   <Text ...
550
550   <Text ...
551
551   <Text ...
552
552   <Text ...
553
553   <Text ...
554
554   <Text ...
555
555   <Text ...
556
556   <Text ...
557
557   <Text ...
558
558   <Text ...
559
559   <Text ...
560
560   <Text ...
561
561   <Text ...
562
562   <Text ...
563
563   <Text ...
564
564   <Text ...
565
565   <Text ...
566
566   <Text ...
567
567   <Text ...
568
568   <Text ...
569
569   <Text ...
570
570   <Text ...
571
571   <Text ...
572
572   <Text ...
573
573   <Text ...
574
574   <Text ...
575
575   <Text ...
576
576   <Text ...
577
577   <Text ...
578
578   <Text ...
579
579   <Text ...
580
580   <Text ...
581
581   <Text ...
582
582   <Text ...
583
583   <Text ...
584
584   <Text ...
585
585   <Text ...
586
586   <Text ...
587
587   <Text ...
588
588   <Text ...
589
589   <Text ...
590
590   <Text ...
591
591   <Text ...
592
592   <Text ...
593
593   <Text ...
594
594   <Text ...
595
595   <Text ...
596
596   <Text ...
597
597   <Text ...
598
598   <Text ...
599
599   <Text ...
600
600   <Text ...
601
601   <Text ...
602
602   <Text ...
603
603   <Text ...
604
604   <Text ...
605
605   <Text ...
606
606   <Text ...
607
607   <Text ...
608
608   <Text ...
609
609   <Text ...
610
610   <Text ...
611
611   <Text ...
612
612   <Text ...
613
613   <Text ...
614
614   <Text ...
615
615   <Text ...
616
616   <Text ...
617
617   <Text ...
618
618   <Text ...
619
619   <Text ...
620
620   <Text ...
621
621   <Text ...
622
622   <Text ...
623
623   <Text ...
624
624   <Text ...
625
625   <Text ...
626
626   <Text ...
627
627   <Text ...
628
628   <Text ...
629
629   <Text ...
630
630   <Text ...
631
631   <Text ...
632
632   <Text ...
633
633   <Text ...
634
634   <Text ...
635
635   <Text ...
636
636   <Text ...
637
637   <Text ...
638
638   <Text ...
639
639   <Text ...
640
640   <Text ...
641
641   <Text ...
642
642   <Text ...
643
643   <Text ...
644
644   <Text ...
645
645   <Text ...
646
646   <Text ...
647
647   <Text ...
648
648   <Text ...
649
649   <Text ...
650
650   <Text ...
651
651   <Text ...
652
652   <Text ...
653
653   <Text ...
654
654   <Text ...
655
655   <Text ...
656
656   <Text ...
657
657   <Text ...
658
658   <Text ...
659
659   <Text ...
660
660   <Text ...
661
661   <Text ...
662
662   <Text ...
663
663   <Text ...
664
664   <Text ...
665
665   <Text ...
666
666   <Text ...
667
667   <Text ...
668
668   <Text ...
669
669   <Text ...
670
670   <Text ...
671
671   <Text ...
672
672   <Text ...
673
673   <Text ...
674
674   <Text ...
675
675   <Text ...
676
676   <Text ...
677
677   <Text ...
678
678   <Text ...
679
679   <Text ...
680
680   <Text ...
681
681   <Text ...
682
682   <Text ...
683
683   <Text ...
684
684   <Text ...
685
685   <Text ...
686
686   <Text ...
687
687   <Text ...
688
688   <Text ...
689
689   <Text ...
690
690   <Text ...
691
691   <Text ...
692
692   <Text ...
693
693   <Text ...
694
694   <Text ...
695
695   <Text ...
696
696   <Text ...
697
697   <Text ...
698
698   <Text ...
699
699   <Text ...
700
700   <Text ...
701
701   <Text ...
702
702   <Text ...
703
703   <Text ...
704
704   <Text ...
705
705   <Text ...
706
706   <Text ...
707
707   <Text ...
708
708   <Text ...
709
709   <Text ...
710
710   <Text ...
711
711   <Text ...
712
712   <Text ...
713
713   <Text ...
714
714   <Text ...
715
715   <Text ...
716
716   <Text ...
717
717   <Text ...
718
718   <Text ...
719
719   <Text ...
720
720   <Text ...
721
721   <Text ...
722
722   <Text ...
723
723   <Text ...
724
724   <Text ...
725
725   <Text ...
726
726   <Text ...
727
727   <Text ...
728
728   <Text ...
729
729   <Text ...
730
730   <Text ...
731
731   <Text ...
732
732   <Text ...
733
733   <Text ...
734
734   <Text ...
735
735   <Text ...
736
736   <Text ...
737
737   <Text ...
738
738   <Text ...
739
739   <Text ...
740
740   <Text ...
741
741   <Text ...
742
742   <Text ...
743
743   <Text ...
744
744   <Text ...
745
745   <Text ...
746
746   <Text ...
747
747   <Text ...
748
748   <Text ...
749
749   <Text ...
750
750   <Text ...
751
751   <Text ...
752
752   <Text ...
753
753   <Text ...
754
754   <Text ...
755
755   <Text ...
756
756   <Text ...
757
757   <Text ...
758
758   <Text ...
759
759   <Text ...
760
760   <Text ...
761
761   <Text ...
762
762   <Text ...
763
763   <Text ...
764
764   <Text ...
765
765   <Text ...
766
766   <Text ...
767
767   <Text ...
768
768   <Text ...
769
769   <Text ...
770
770   <Text ...
771
771   <Text ...
772
772   <Text ...
773
773   <Text ...
774
774   <Text ...
775
775   <Text ...
776
776   <Text ...
777
777   <Text ...
778
778   <Text ...
779
779   <Text ...
780
780   <Text ...
781
781   <Text ...
782
782   <Text ...
783
783   <Text ...
784
784   <Text ...
785
785   <Text ...
786
786   <Text ...
787
787   <Text ...
788
788   <Text ...
789
789   <Text ...
790
790   <Text ...
791
791   <Text ...
792
792   <Text ...
793
793   <Text ...
794
794   <Text ...
795
795   <Text ...
796
796   <Text ...
797
797   <Text ...
798
798   <Text ...
799
799   <Text ...
800
800   <Text ...
801
801   <Text ...
802
802   <Text ...
803
803   <Text ...
804
804   <Text ...
805
805   <Text ...
806
806   <Text ...
807
807   <Text ...
808
808   <Text ...
809
809   <Text ...
810
810   <Text ...
811
811   <Text ...
812
812   <Text ...
813
813   <Text ...
814
814   <Text ...
815
815   <Text ...
816
816   <Text ...
817
817   <Text ...
818
818   <Text ...
819
819   <Text ...
820
820   <Text ...
821
821   <Text ...
822
822   <Text ...
823
823   <Text ...
824
824   <Text ...
825
825   <Text ...
826
826   <Text ...
827
827   <Text ...
828
828   <Text ...
829
829   <Text ...
830
830   <Text ...
831
831   <Text ...
832
832   <Text ...
833
833   <Text ...
834
834   <Text ...
835
835   <Text ...
836
836   <Text ...
837
837   <Text ...
838
838   <Text ...
839
839   <Text ...
840
840   <Text ...
841
841   <Text ...
842
842   <Text ...
843
843   <Text ...
844
844   <Text ...
845
845   <Text ...
846
846   <Text ...
847
847   <Text ...
848
848   <Text ...
849
849   <Text ...
850
850   <Text ...
851
851   <Text ...
852
852   <Text ...
853
853   <Text ...
854
854   <Text ...
855
855   <Text ...
856
856   <Text ...
857
857   <Text ...
858
858   <Text ...
859
859   <Text ...
860
860   <Text ...
861
861   <Text ...
862
862   <Text ...
863
863   <Text ...
864
864   <Text ...
865
865   <Text ...
866
866   <Text ...
867
867   <Text ...
868
868   <Text ...
869
869   <Text ...
870
870   <Text ...
871
871   <Text ...
872
872   <Text ...
873
873   <Text ...
874
874   <Text ...
875
875   <Text ...
876
876   <Text ...
877
877   <Text ...
878
878   <Text ...
879
879   <Text ...
880
880   <Text ...
881
881   <Text ...
882
882   <Text ...
883
883   <Text ...
884
884   <Text ...
885
885   <Text ...
886
886   <Text ...
887
887   <Text ...
888
888   <Text ...
889
889   <Text ...
890
890   <Text ...
891
891   <Text ...
892
892   <Text ...
893
893   <Text ...
894
894   <Text ...
895
895   <Text ...
896
896   <Text ...
897
897   <Text ...
898
898   <Text ...
899
899   <Text ...
900
900   <Text ...
901
901   <Text ...
902
902   <Text ...
903
903   <Text ...
904
904   <Text ...
905
905   <Text ...
906
906   <Text ...
907
907   <Text ...
908
908   <Text ...
909
909   <Text ...
910
910   <Text ...
911
911   <Text ...
912
912   <Text ...
913
913   <Text ...
914
914   <Text ...
915
915   <Text ...
916
916   <Text ...
917
917   <Text ...
918
918   <Text ...
919
919   <Text ...
920
920   <Text ...
921
921   <Text ...
922
922   <Text ...
923
923   <Text ...
924
924   <Text ...
925
925   <Text ...
926
926   <Text ...
927
927   <Text ...
928
928   <Text ...
929
929   <Text ...
930
930   <Text ...
931
931   <Text ...
932
932   <Text ...
933
933   <Text ...
934
934   <Text ...
935
935   <Text ...
936
936   <Text ...
937
937   <Text ...
938
938   <Text ...
939
939   <Text ...
940
940   <Text ...
941
941   <Text ...
942
942   <Text ...
943
943   <Text ...
944
944   <Text ...
945
945   <Text ...
946
946   <Text ...
947
947   <Text ...
948
948   <Text ...
949
949   <Text ...
950
950   <Text ...
951
951   <Text ...
952
952   <Text ...
953
953   <Text ...
954
954   <Text ...
955
955   <Text ...
956
956   <Text ...
957
957   <Text ...
958
958   <Text ...
959
959   <Text ...
960
960   <Text ...
961
961   <Text ...
962
962   <Text ...
963
963   <Text ...
964
964   <Text ...
965
965   <Text ...
966
966   <Text ...
967
967   <Text ...
968
968   <Text ...
969
969   <Text ...
970
970   <Text ...
971
971  
```

Sprites



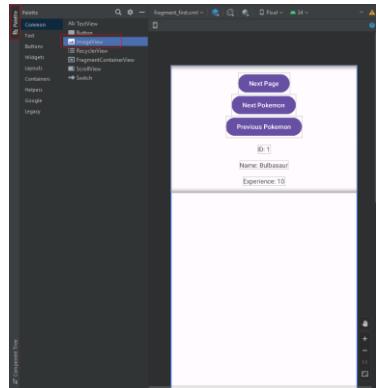
```
13763     "name": "charmander",
13764     "url": "https://pokeapi.co/api/v2/pokemon-species/4/"
13765   },
13766   "sprites": {
13767     "back_default": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/back/4.png",
13768     "back_female": null,
13769     "back_shiny": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/back/shiny/4.png",
13770     "back_shiny_female": null,
13771     "front_default": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/4.png",
13772     "front_female": null,
13773     "front_shiny": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/shiny/4.png",
13774     "front_shiny_female": null,
13775   },
13776   "other": {
13777     "dream_world": {
13778       "front_default": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/dream-world/4.svg",
13779       "front_female": null
13780     },
13781     "home": {
13782       "front_default": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/home/4.png",
13783       "front_female": null,
13784       "front_shiny": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/home/shiny/4.png",
13785       "front_shiny_female": null
13786     },
13787     "official-artwork": {
13788       "front_default": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/4.png",
13789       "front_shiny": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/shiny/4.png"
13790     }
13791   }
13792 }
```

Bei Pokemon Klasse String ergänzt

```
public String imageUrlDreamWorld = "";
```

Im Layout Image View rein (Über die Palette) => GEHT NICHT

⇒ Aber wir wissen, es muss „ImageView“ heißen



```
<ImageView
    android:id="@+id/pokemon_image"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_gravity="center"
    android:layout_marginTop="16dp"
    android:background="@color/black"/>
```

Im FirstFragment machen wir das Binding zum Layout:

```
binding.pokemonImage.setImageBitmap();
```

.setImageBitmap() setzt Bild ohne compression

Als nächstes in der PokeAPI / readPokemon() das ergänzen

```
case "sprites": //Darunter weitere geschungene Klammer, daher neues BeginObjekt

    reader.beginObject(); //BeginObjekt hat immer Reader.hasNext()
    while (reader.hasNext()) {
        switch (reader.nextName()) {
            case "other": //Darunter weitere geschungene Klammer, daher neues BeginObjekt

                reader.beginObject();
                while (reader.hasNext()) {
                    switch (reader.nextName()) {
                        case "dream_world": //Darunter weitere geschungene Klammer, daher neues BeginObjekt

                            reader.beginObject();
                            while (reader.hasNext()) {
                                switch (reader.nextName()) {
                                    case "front_default":
                                        pokemon.imageUrlDreamWorld = reader.nextString();
                                        break; // Weil ganz unten, bitte verlasse das switch
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Dahinter muss jetzt überall noch nen break, Weil ganz unten, bitte verlasse das switch und dann der nächste drüber etc.

```
break;

case "sprites": //Darunter weitere geschungene Klammer, daher neues BeginObjekt

    reader.beginObject(); //BeginObjekt hat immer Reader.hasNext()
    while (reader.hasNext()) {
        switch (reader.nextName()) {
            case "other": //Darunter weitere geschungene Klammer, daher neues BeginObjekt

                reader.beginObject();
                while (reader.hasNext()) {
                    switch (reader.nextName()) {
                        case "dream_world": //Darunter weitere geschungene Klammer, daher neues BeginObjekt

                            reader.beginObject();
                            while (reader.hasNext()) {
                                switch (reader.nextName()) {
                                    case "front_default":
                                        pokemon.imageUrlDreamWorld = reader.nextString();
                                        break; // Weil ganz unten, bitte verlasse das switch
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

default:
    reader.skipValue();
}
```

Formatierung mit Strg+Alt+O / Strg+Alt+L

Als nächstes brauchen wir noch

```
default:  
    reader.skipValue();
```

überall – Werte die wir nicht kennen werden übersprungen

Als nächstes brauchen wir noch das Ende jeder while Schleife

```
reader.endObject();
```

Objekt muss geschlossen werden

```
case "sprites": //Darunter weitere geschungene Klammer, daher neues BeginObjekt  
  
    reader.beginObject(); //BeginObjekt hat immer Reader.hasNext  
    while (reader.hasNext()) {  
        switch (reader.nextName()) {  
            case "other": //Darunter weitere geschungene Klammer, daher neues BeginObjekt  
  
                reader.beginObject();  
                while (reader.hasNext()) {  
                    switch (reader.nextName()) {  
                        case "dream_world": //Darunter weitere geschungene Klammer, daher neues BeginObjekt  
  
                            reader.beginObject();  
                            while (reader.hasNext()) {  
                                switch (reader.nextName()) {  
                                    case "front_default":  
                                        pokemon.imageUrlDreamWorld = reader.nextString();  
                                        break; // Weil ganz unten, bitte verlasse das switch  
                                    default:  
                                        reader.skipValue();  
                                }  
                            }  
                            reader.endObject();  
                            break;  
                        default:  
                            reader.skipValue();  
                    }  
                }  
                reader.endObject();  
                break;  
            default:  
                reader.skipValue();  
        }  
    }  
    reader.endObject();  
    break;  
default:
```

Ganz unten müssen wir jetzt noch schauen, ob ein Wert null ist

```
if (reader.peek() == JsonToken.STRING) {  
    pokemon.imageUrlDreamWorld = reader.nextString();  
} else reader.skipValue();
```

```

        case "sprites": //Darunter weitere geschungene Klammer, daher neues BeginObjekt

            reader.beginObject(); //BeginObjekt hat immer Reader.hasNext
            while (reader.hasNext()) {
                switch (reader.nextName()) {
                    case "other": //Darunter weitere geschungene Klammer, daher neues BeginObjekt

                        reader.beginObject();
                        while (reader.hasNext()) {
                            switch (reader.nextName()) {
                                case "dream_world": //Darunter weitere geschungene Klammer, daher neues BeginObjekt

                                    reader.beginObject();
                                    while (reader.hasNext()) {
                                        switch (reader.nextName()) {
                                            case "front_default":
                                                if (reader.peek() == JsonToken.STRING) {
                                                    pokemon.imageUrlDreamWorld = reader.nextString();
                                                } else reader.skipValue();
                                                break; // Weil ganz unten, bitte verlasse das switch
                                            default:
                                                reader.skipValue();
                                        }
                                    }
                                    reader.endObject();
                                    break;
                                default:
                                    reader.skipValue();
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Ergebnis:

Im FirstFragment fragen wir nach PokeAPI und fragen nach einem Pokemon. In dem Pokemon wurde die Variable „imageUrlDreamWorld“ ergänzt

Im First Fragment haben wir Binding zum Layout ergänzt mit Funktion setImageBitmap()

Wir haben jetzt nur die URL. Die Bild-Daten müssen noch aus dem Internet runtergeladen werden.

```

//2
2 usages ▲ Isa
JsonReader requestJsonReader(String url) throws IOException { //JSON lesen
    HttpURLConnection connection = (HttpURLConnection) new URL(url).openConnection();
    connection.setRequestProperty("User-Agent", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Sa
    BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(connection.getInputStream(), charsetName: "UTF-8")); //Datenstrom als UTF-
    return new JsonReader(bufferedReader); //Text/JSON als Antwort
}

```

Hiermit lesen wir die Daten aus dem Internet. Jetzt lesen wir kein JSON sondern nach dem JSON zusätzlich noch ein Bild

URL zu BILD

Neu 04.12.23

Bitmap Reader bauen um Bilder auszulesen

In PokeAPI ImageLoader Klasse erstellt

```

public static class ImageLoader {

    public static Bitmap loadImageFromUrl(String imageUrl) throws IOException {
        HttpURLConnection connection = (HttpURLConnection) new URL(imageUrl).openConnection();
        connection.setRequestProperty("User-Agent", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36");

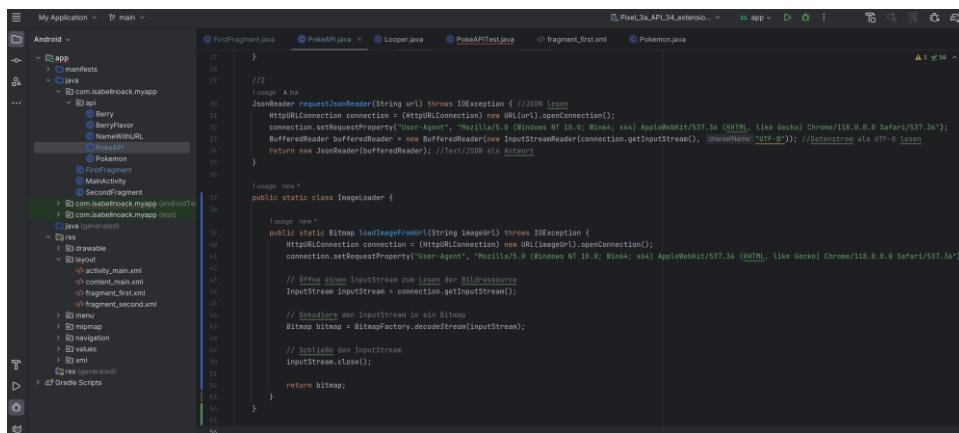
        // Öffne einen InputStream zum Lesen der Bildressource
        InputStream inputStream = connection.getInputStream();

        // Dekodieere den InputStream in ein Bitmap
        Bitmap bitmap = BitmapFactory.decodeStream(inputStream);

        // Schließe den InputStream
        inputStream.close();

        return bitmap;
    }
}

```



Dann noch im FirstFragment.java das binding ergänzt

```

// Bild laden und in ImageView setzen
try {
    Bitmap bitmap = PokeAPI.ImageLoader.loadImageFromUrl(pokemon.imageUrlDreamWorld); // Hier wird der
    entsprechenden Bild-URL-Pfad übergeben
    binding.pokemonImage.setImageBitmap(bitmap); // an Layout übergeben
} catch (IOException e) {
    e.printStackTrace(); // Fehlermeldung
}

```

```

    76     @SuppressWarnings("SetTextI18n") //Nicht hervorheben
    77     void loadPokemon() {
    78         //Neuer Thread da Hauptthread nicht blockiert werden darf
    79         new Thread(() -> {
    80             try {
    81                 Pokemon pokemon = new PokeAPI().requestPokemon(pokemonId); //Funktion der neuen Instanz pokeAPI aufrufen
    82                 getActivity().runOnUiThread() -> {
    83                     //UI
    84                     String name = pokemon.name;
    85                     name = Character.toUpperCase(name.charAt(0)) + name.substring(1); //Erster Charakter groß geschrieben
    86                     binding.pokemonName.setText(name);
    87                     binding.pokemonExperience.setText("Experience: " + pokemon.baseExperience);
    88                     binding.pokemonId.setText("ID: " + pokemonId);
    89
    90                     // Bild laden und in ImageView setzen
    91                     try {
    92                         Bitmap bitmap = PokeAPI.ImageLoader.loadImageFromUrl(pokemon.imageUrlDreamWorld); // Hier wird der entsprechenden Bild-URL-Pfad übergeben
    93                         binding.pokemonImage.setImageBitmap(bitmap); //Hier wird das entsprechende Bild übergeben
    94                     } catch (IOException e) {
    95                         e.printStackTrace(); //Fehlermeldung
    96                     }
    97                 });
    98             } catch (IOException ignored) {
    99                 // todo send warning to user
    100            }
    101        }).start();
    102    }

```

Hier ist der Fehler

Android.os.NetworkOnMainThreadException

Das heisst, es muss wieder auf einem eigenen Thread laufen!!

Daher um das drumrum nochmal neuer Thread bauen:

```

    // Bild laden und in ImageView setzen
    try {
        new *
        new Thread(new Runnable() {
            new *
            @Override
            public void run() {
                try {
                    final Bitmap bitmap = PokeAPI.ImageLoader.loadImageFromUrl(pokemon.imageUrlDreamWorld);

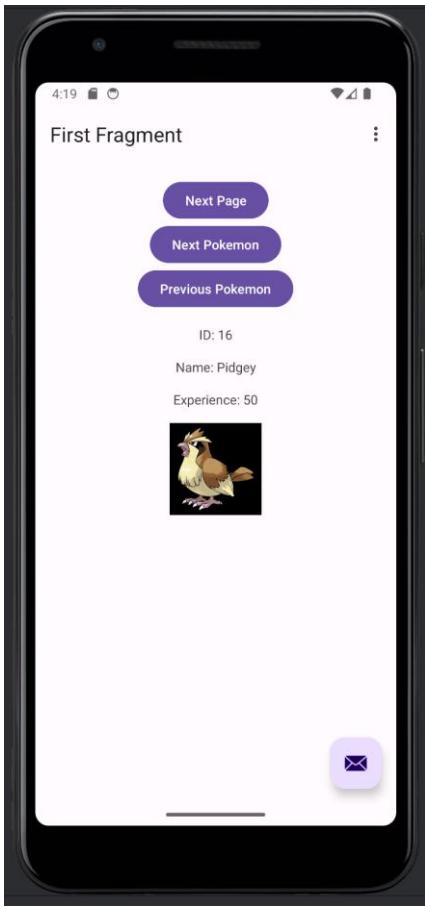
                    new *
                    getActivity().runOnUiThread(new Runnable() {
                        new *
                        @Override
                        public void run() {
                            binding.pokemonImage.setImageBitmap(bitmap);
                        }
                    });
                } catch (IOException e) {
                    e.printStackTrace(); //Fehlermeldung
                }
            }
        }).start();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Bild wird nicht angezeigt weil ich die Sprites von Dream World nehmen wollte, die sind aber leider .svg was der ImageLoader nicht kann...

Hab stattdessen die “official-artwork” genommen, weil das png sind 😊 - Erfolg!

Zusätzlich sind die official artworks vollständig, die dream world bilder haben manchmal null



Neu 18.12.23

Pokemon Liste bauen

dafür Fragmente neu strukturieren

pokemon first fragment muss aus seite 2 kommen, auf seite first fragment dann die liste mit pokemon

Umbenennungen:

First Fragment zu PokemonFragment.java

SecondFragment zu ListFragment.java

fragment_first.xml zu fragment_pokemon.xml

fragment_second.xml zu fragment_list.xml

FragmentFirstBinding zu FragmentPokemonBinding

FragmentSecondBinding zu FragmentListBinding

ID Umbenennung im nav_graph

Seiten-Bezeichnung Umbenennung im strings.xml

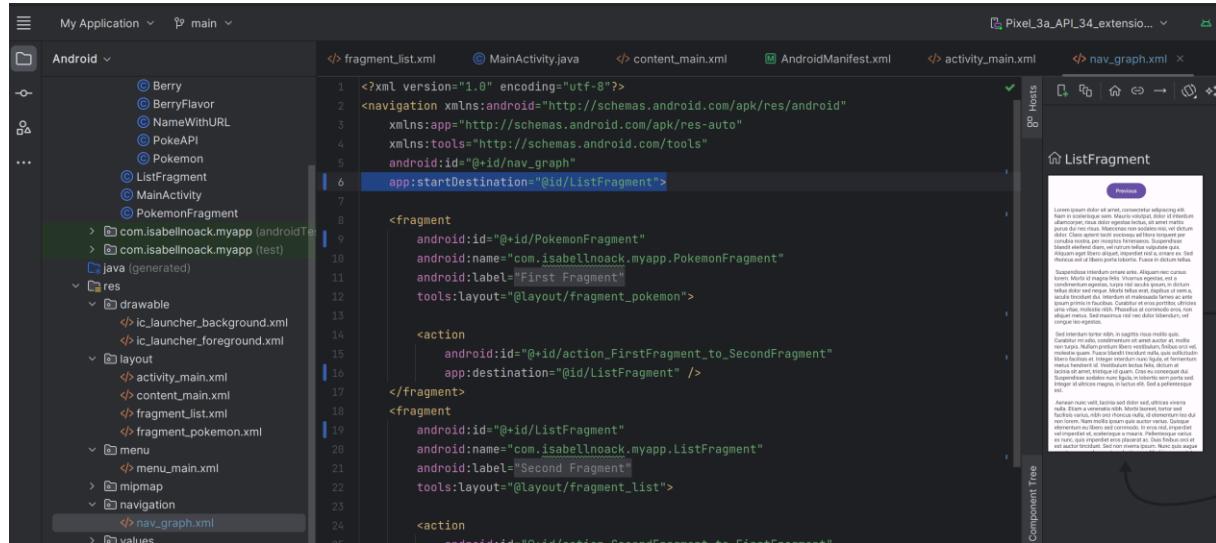
Appstart mit ListFragment

=> in nav_graph.xml

```
app:startDestination="@+id/PokemonFragment">
```

Umgeändert zu

```
app:startDestination="@+id/ListFragment">
```



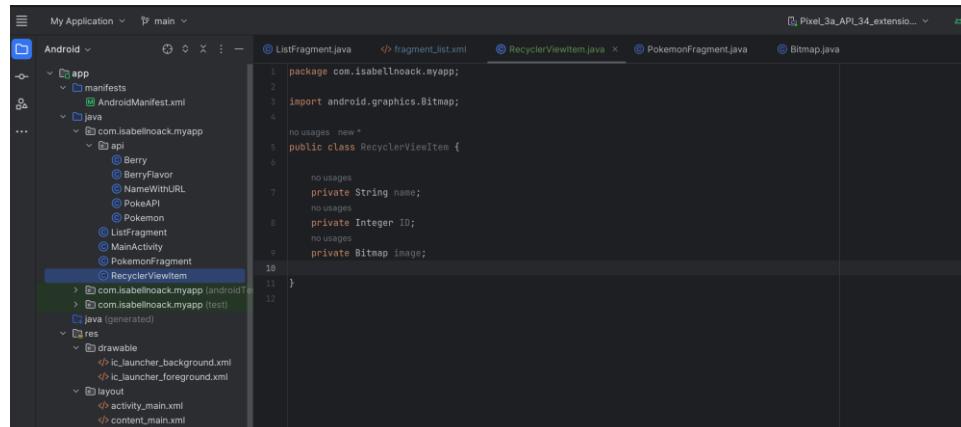
Liste mit Pokemon Bildern

anhand dem Tutorial <https://www.youtube.com/watch?v=jTo8aYcd-m8>



=> in Layout (fragment_list.xml) RecyclerView erstellt

Neue Klasse: RecyclerviewItem

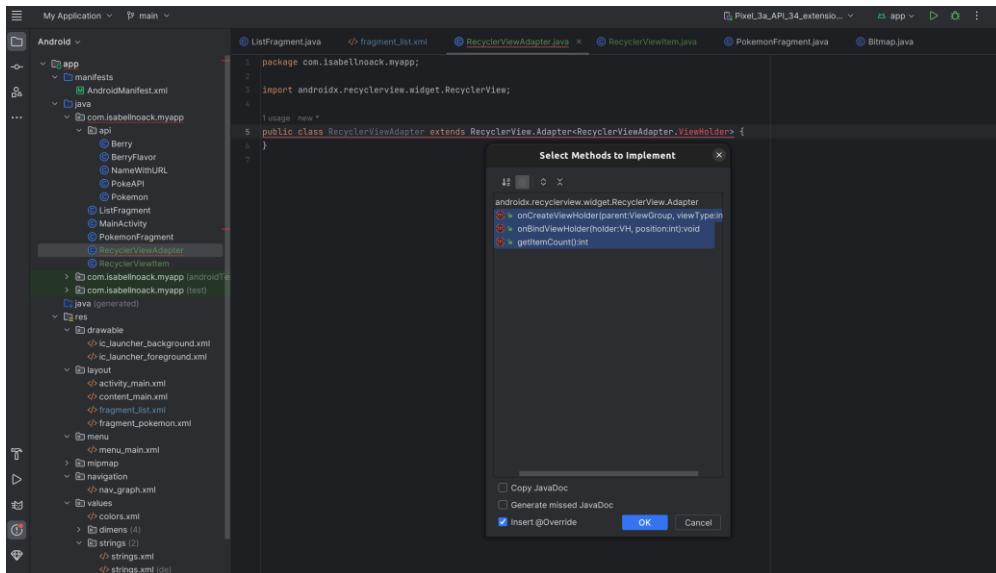


Klasse angelegt: RecyclerViewAdapter

Problem: Eine RecyclerList kann kein zeug einfach anzeigen, daher braucht man den Adapter...

Zusammengefasst: Der Adapter bindet Daten an eine RecyclerView, indem er die Daten in ViewHolders lädt, die dann in der RecyclerView angezeigt werden. Der ViewHolder enthält Ansichten für jedes Element, und der Adapter kümmert sich um das Handling der Daten und deren Darstellung.

Klasse RecyclerViewAdapter gebaut (extends RecyclerViewAdapter)



```
3 usages new *
public class ViewHolder extends RecyclerView.ViewHolder {
```

no usages new *
public ViewHolder(@NonNull View itemView) {
 super(itemView);
}

```
}
```

```
3 usages new *
public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
```

no usages
💡 private ArrayList<RecyclerViewItem> recyclerViewItems; //ArrayList<typ> variablen-name

klasse ViewHolder gebaut (extends RecyclerView.ViewHolder)

Danach noch ArrayList aus den Items

Wie viele Items ich in meiner RecyclerView habe

```

new *

@Override
public int getItemCount() {
    return recyclerViewItems.size();
}

```

Verstehe ich nicht:

```

1 usage
private ArrayList<RecyclerViewItem> recyclerViewItems; //ArrayList<typ> variablen-name

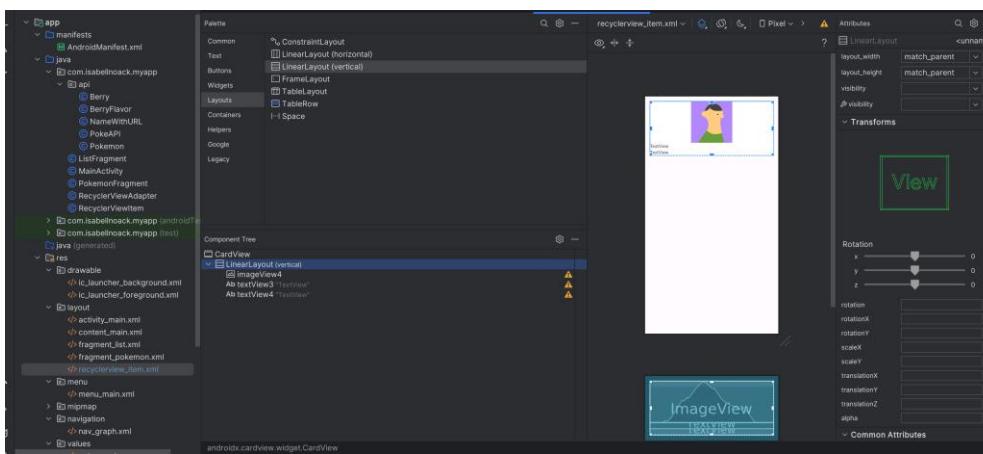
//für was wird das verwendet?
no usages
private Context context;

new *
@NonNull
@Override

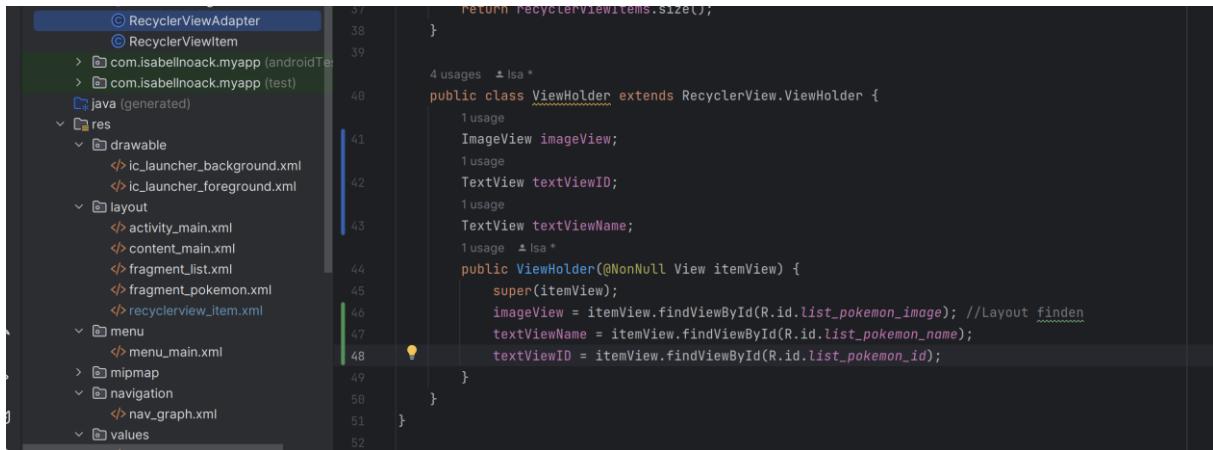
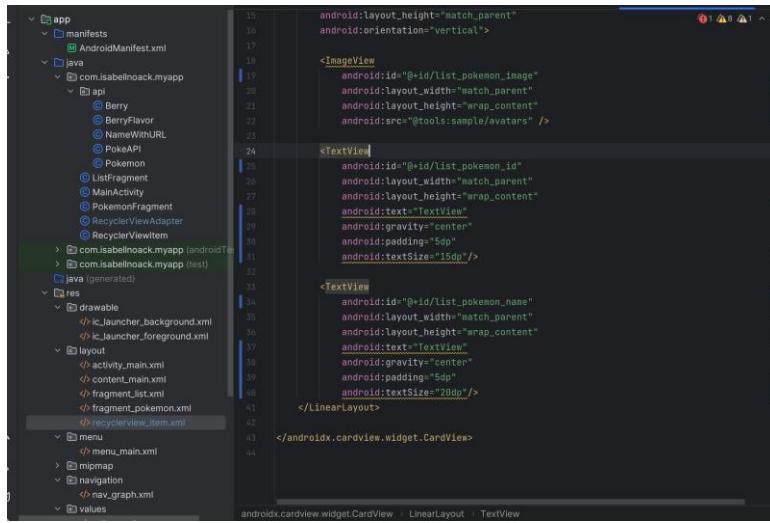
public RecyclerViewAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.recyclerview_item,parent, attachToRoot: false);
    return new ViewHolder(view);
}

```

Neues Layout für die Items erstellt:
recyclerview_item.xml



Das in der Klasse verknüpfen mit der jeweiligen ID (Funktion: findViewById)
für jedes der Elemente in meinem Layout



ausserdem den Holder erstellen für jedes der Elemente

und anderer kram den ich nicht versteh

- ListFragment versteh ich nicht und das RecyclerViewItem und wie das alles zusammen kommt

Liste mit Pokemon Bildern – 1,2, oder 3 Spalten nebeneinander

im ListFragment.java

```

//Hier wird der GridLayoutManager mit 3 Spalten erstellt
int numberofColumns = 3;
GridLayoutManager layoutManager = new GridLayoutManager(context, numberofColumns);
binding.recyclerView.setLayoutManager(layoutManager);

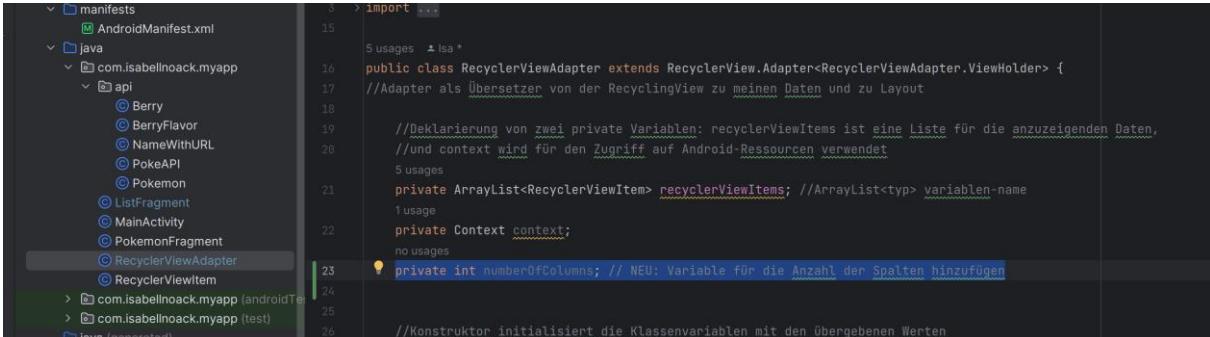
```

=> jetzt muss die grösse der Bilder und Texte angepasst werden

=> erstmal wenn 1 Column, dann soll die textView nebeneinander sein und ansonsten unter einander

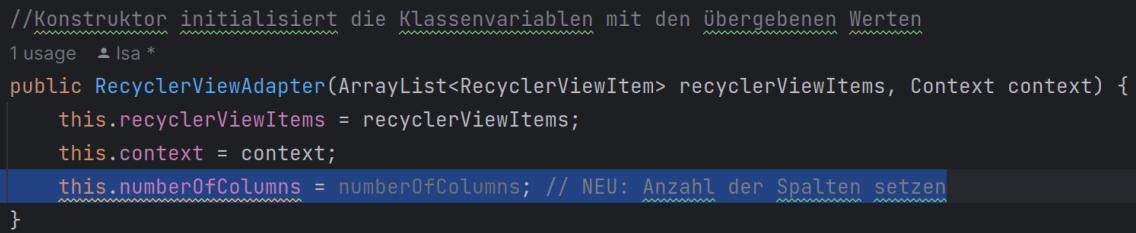
Im bestehenden RecyclerViewAdapter-CodeAnpassungen vornehmen, um die Anordnung der TextViews basierend auf der Anzahl der Spalten zu ändern.

Alles im RecyclerViewAdapter!



```
import android.widget.TextView;
import java.util.ArrayList;
import java.util.List;

public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
    // Adapter als Übersetzer von der RecyclerView zu meinen Daten und zu Layout
    // Deklarierung von zwei private Variablen: recyclerViewItems ist eine Liste für die anzugegenden Daten,
    // und context wird für den Zugriff auf Android-Ressourcen verwendet
    private ArrayList<RecyclerViewItem> recyclerViewItems; //ArrayList<typ> variablen-name
    private Context context;
    private int numberOfColumns; // NEU: Variable für die Anzahl der Spalten hinzufügen
    // Konstruktor initialisiert die Klassenvariablen mit den übergebenen Werten
}
```



```
//Konstruktor initialisiert die Klassenvariablen mit den übergebenen Werten
public RecyclerViewAdapter(ArrayList<RecyclerViewItem> recyclerViewItems, Context context) {
    this.recyclerViewItems = recyclerViewItems;
    this.context = context;
    this.numberOfColumns = numberOfColumns; // NEU: Anzahl der Spalten setzen
}
```

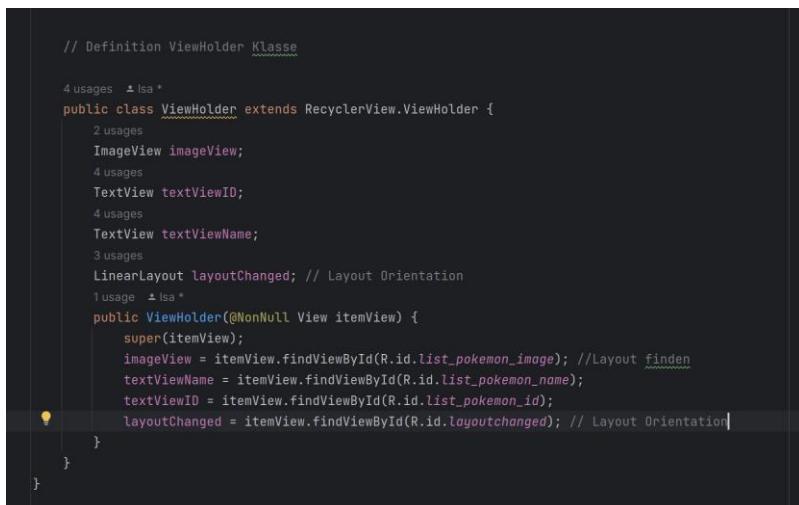
Die TextViews finden wir durch unsere eigene Definition, wie die ViewHolder Klasse aussieht:

textViewName und textViewID wird hier zu den Ressourcen (layout) vernetzt

Was wir alles ändern müssen um von 1 Spalte zu 2 oder mehr zu gehen:

- Layout orientation von horizontal zu vertical
- in den TextViews die android:layout_width="0dp" zu ="match_parent"

Für das Layout: Dafür auch im ViewHolder das Layout connected über die ID layoutchanged



```
// Definition ViewHolder Klasse
public class ViewHolder extends RecyclerView.ViewHolder {
    ImageView imageView;
    TextView textViewID;
    TextView textViewName;
    LinearLayout layoutChanged; // Layout Orientation
    public ViewHolder(@NotNull View itemView) {
        super(itemView);
        imageView = itemView.findViewById(R.id.list_pokemon_image); //Layout finden
        textViewName = itemView.findViewById(R.id.list_pokemon_name);
        textViewID = itemView.findViewById(R.id.list_pokemon_id);
        layoutChanged = itemView.findViewById(R.id.layoutchanged); // Layout Orientation
    }
}
```

```

53
54
55
56
57
58
59
60
61
62
63
64
65
66
67

```

```

    // Anpassung der Anordnung der TextViews basierend auf der Anzahl der Spalten
    if (numberOfColumns == 1) {
        // Wenn es nur eine Spalte gibt, setze die TextViews nebeneinander und das Layout horizontal
        holder.textViewName.getLayoutParams().width = 0; // layout_width auf 0dp setzen
        holder.textViewID.getLayoutParams().width = 0; // layout_width auf 0dp setzen
        holder.layoutChanged.setOrientation(LinearLayout.HORIZONTAL);
    } else {
        // Wenn es mehr als eine Spalte gibt, setze die TextViews untereinander und das Layout vertikal
        holder.textViewName.getLayoutParams().width = ViewGroup.LayoutParams.MATCH_PARENT; // layout_width auf match_parent setzen
        holder.textViewID.getLayoutParams().width = ViewGroup.LayoutParams.MATCH_PARENT; // layout_width auf match_parent setzen
        holder.layoutChanged.setOrientation(LinearLayout.VERTICAL);
    }
}

```

Liste mit allen Pokemon – Button erstellen für die Spalten-Änderung

Button erstellen, welches den parameter ändert. Danach muss die Liste geupdated werden.

```

<Button
    android:id="@+id/button_numberOfColumns"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Columns"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/button_search"
    app:layout_constraintTop_toTopOf="parent"/>

```

Wenn Button geklickt wird:

```

//Button Columns
new *
binding.buttonNumberOfColumns.setOnClickListener(new View.OnClickListener() {
    new *
    @Override
    public void onClick(View view) {
        if (numberOfColumns == 3) {
            numberOfColumns = 1;
        } else {
            numberOfColumns++;
        }
        updateRecyclerView();
    }
});

```

dafür die Funktion updateRecyclerView(); erstellt!

```

// Methode zur Aktualisierung der RecyclerView mit der neuen Anzahl von Spalten
@Usage(new = true)
private void updateRecyclerView() {
    GridLayoutManager layoutManager = new GridLayoutManager(getContext(), number_of_columns);
    binding.recyclerView.setLayoutManager(layoutManager);

    // Adapter für die RecyclerView erneut setzen
    RecyclerViewAdapter recyclerViewAdapter = new RecyclerViewAdapter(recyclerViewItems, getContext(), number_of_columns);
    binding.recyclerView.setAdapter(recyclerViewAdapter);
}

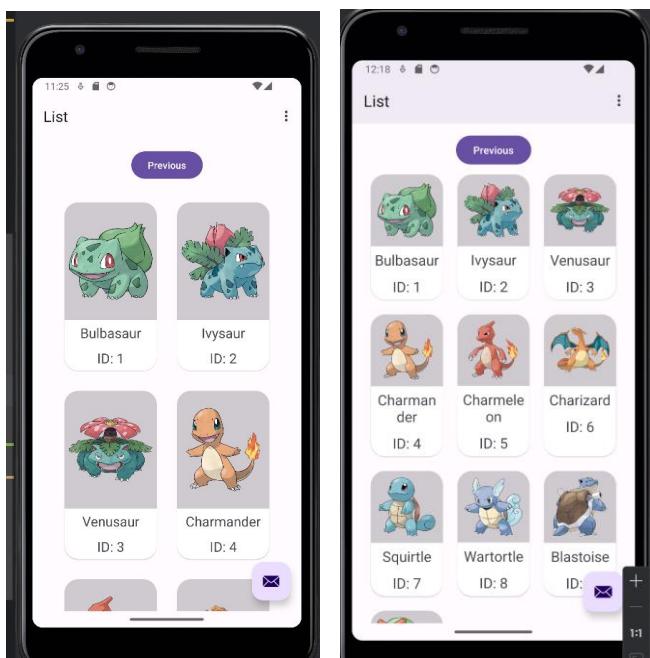
```

Funktioniert alles wie gewollt, durch drücken des Buttons „Columns“ wird zwischen 1,2 und 3 Spalten gewechselt.



Liste mit allen Pokemon – mit Schleife

```
//Schleife, Listen Eintrag für alle Pokemon ID von 1 bis 1017
for (int id = 1; id <= 10; id++) {
    recyclerViewItems.add(new RecyclerViewItem(id)); //Listen Eintrag
}
//To-Do: performantes laden!
```



Höhe noch angepasst im XML

```
android:adjustViewBounds="true"
```

Zu grosse Pokemon Abfrage auf einmal

Problem: wenn ich alle 1017 Pokemon vorlade – braucht die App mehrere Minuten bis etwas angezeigt wird!

Bei 100 Pokemon hat es bereits 40sek gedauert.

=> Cashing oder offset und limit stuff?

Stattdessen Code umgebaut:

Man hat keine ArrayListe „recyclerViewItems“ mehr, die alle Pokemon an RecyclerViewAdapter gibt.

Jetzt machen wir es so, dass der Adapter sich selber zieht was er braucht. Damit sollte er nur das anfragen, was man auf dem Handy sieht. Und nicht alles auf einmal.

Das Laden ist jetzt im Adapter (onBindViewHolder) statt ListFragment

=> ArrayList (aus dem listFragment) gelöscht und statt dass wir das bei erstellen von dem ViewHolder übergeben, ist das jetzt direkt im ViewHolder drinnen

```

75         //Array Liste neues Item hinzufügen
    -     try {
    -         //Schleife, Listen-Eintrag für alle Pokemon ID von 1 bis 1017
    -         for (int id = 1; id <= 10; id++) {
    -             recyclerViewItems.add(new RecyclerViewItem(id)); //Listen Eintrag
    -         }
    -         //To-Do: performantes laden!
    -
    -     } catch (IOException ignored) {
    -         // todo send warning to user
    -     }

```

das Array recyclerViewItems wird jetzt auch nicht mehr gebraucht

Jetzt neuer Code im RecyclerViewAdapter

```

56 +
57 +     int id = position + 1;
58 +
59 +     Thread t = new Thread(() -> {
60 +         try {
61 +             final RecyclerViewItem item = new RecyclerViewItem(id);
62 +             activity.runOnUiThread(new Runnable() {
63 +                 @Override
64 +                 public void run() {
65 +
66 +                     //holder.imageView.setImageURI(Uri.parse(item.pokemon.imageUrl));
67 +                     holder.imageView.setImageBitmap(item.image);
68 +                     holder.textViewID.setText("ID: " + item.id.toString()); //String mit "ID:" + ID
69 +                     holder.textViewName.setText(item.name);
70 +                 }
71 +             });
72 +
73 +         } catch (IOException e) {
74 +             throw new RuntimeException(e);
75 +         }
76 +     });
77 +
78 +     t.start();

```

Die Pokemon werden jetzt anhand der Position hier mit ID genommen

Code: final RecyclerViewItem item = new RecyclerViewItem(id);

Pokemon IDs 1025 < 1302

Es gibt 1302 Pokemon

Pokemon mit ID 1013 hat kein Sprite!

Pokemon ab ID 1018 haben auch kein Sprite!

Ausserdem sind die IDs ab 1025 im 10.000-Bereich. Das sind anscheinend besondere Formen der Pokemon. Bestimmte entwicklungen, in verschiedenen Regionen etc.

Keine Sprites - Error Behandlung

Pokemon mit ID 1013 hat kein Sprite!

Für das ListFragment:

Im RecyclerViewAdapter:

```
//holder.imageView.setImageURI(Uri.parse(item.pokemon.imageUrl));
if (item.image != null) {
    holder.imageView.setImageBitmap(item.image);
} else {
    holder.imageView.setImageResource(R.drawable.empty);
}
```

statt dem Bild wird das .xml „empty“ gezeigt

Für das PokemonFragment: (Im PokemonFragment)

Wenn das Bild gesetzt wird überprüfen ob es "" hat und wenn, dann empty.xml setzen

```
if(pokemon.imageUrl != "") { //wenn Sprite null > in RecyclerViewItem übersprungen > Standard Wert ""
    final Bitmap bitmap = PokeAPI.ImageLoader.loadImageFromUrl(pokemon.imageUrl);
    getActivity().runOnUiThread(new Runnable() {
        @Override
        public void run() {
            binding.pokemonImage.setImageBitmap(bitmap);
        }
    });
} else { //Hier statt "" dann empty.xml setzen
    binding.pokemonImage.setImageResource(R.drawable.empty);
}
```

empty.xml erstellen aus svg image:

Im ordner drawable rechtsklick => new => Vector Asset => Pfad auswählen

Dokumentation Anmerkung

– **was man danach machen könnte**

- Caching der pokemon
- Favoriten festlegen
- welches sich zu was entwickelt

- Alle Pokemon Entwicklungen rein nehmen (Pokemon IDs ab 10001)
- Sprache wechselbar
- Filterkriterien

App Icon

In der AndroidManifest.xml

```
android:icon="@mipmap/ic_pokeapp"
android:roundIcon="@mipmap/ic_pokeapp"
```

Bild benennung: ic_pokeapp.png (lowercase)

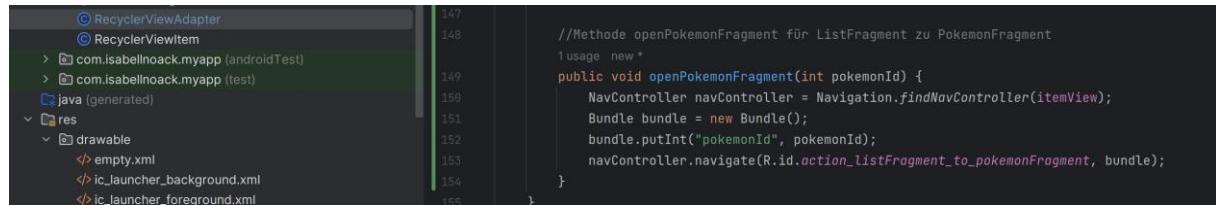
Pokemon Details aufrufen

Von ListFragment zu Pokemon Fragment

Pokemon-ID beim Klicken auf ein RecyclerView-Element an das PokemonFragment übergeben - über die Navigation mit dem NavController

In RecyclerViewAdapter

In der ViewHolder-Klasse Methode hinzufügen (zum Öffnen des PokemonFragments)

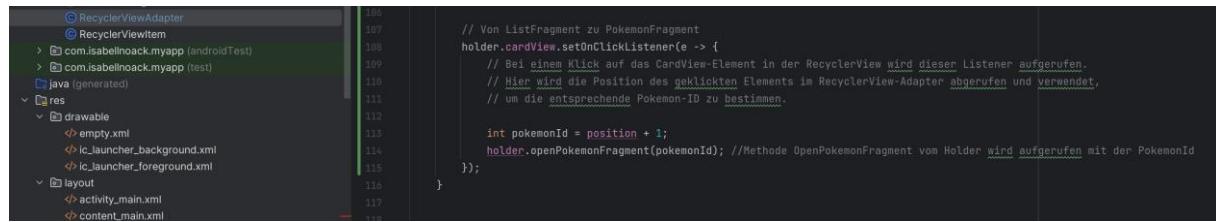


```
147 //Methode openPokemonFragment für ListFragment zu PokemonFragment
148 1usage new "
149 public void openPokemonFragment(int pokemonId) {
150     NavController navController = Navigation.findNavController(itemView);
151     Bundle bundle = new Bundle();
152     bundle.putInt("pokemonId", pokemonId);
153     navController.navigate(R.id.action_listFragment_to_pokemonFragment, bundle);
154 }
155 }
```

In nav_graph.xml action benutzen/umbenannt in:

android:id="@+id/action_listFragment_to_pokemonFragment"

OnClickListener für die CardView



```
109 // Von ListFragment zu PokemonFragment
110 holder.cardView.setOnClickListener(e ->
111     // Bei einem Klick auf das CardView-Element in der RecyclerView wird dieser Listener aufgerufen.
112     // Hier wird die Position des geklickten Elements im RecyclerView-Adapter abgerufen und verwendet,
113     // um die entsprechende Pokemon-ID zu bestimmen.
114
115     int pokemonId = position + 1;
116     holder.openPokemonFragment(pokemonId); //Methode OpenPokemonFragment vom Holder wird aufgerufen mit der PokemonId
117 });
118 }
```

=> Mit Klick von ListFragment zu PokemonFragment funktioniert aber die ID wird nicht richtig übergeben

=> Die Methode openPokemonFragment muss angepasst werden, aber wie?

=> Man kann eine statische Variable für die ganze App festlegen.

```
public static int pokémonIdToOpen;
```

- public: von überall aus dem Programm zugänglich
- static: Statische Methoden können aufgerufen werden, ohne dass eine Instanz der Klasse erstellt werden muss;
- int: Datentyp

Variable definiert in der MainActivity.java

```
public static int pokémonIdToOpen;
```

RecyclerViweAdapter angepasst mit der neuen Variable

```
holder.cardView.setOnClickListener(e -> {
    pokémonIdToOpen = position + 1;
    holder.openPokemonFragment(pokémonIdToOpen);
});
```

Im PokemonFragment die pokémonId zu *pokémonIdToOpen* anstatt festem Startwert

```
int pokémonId = pokémonIdToOpen;
```

Aber Problem für später: Der Wert pokémonIdToOpen kommt aus der CardView Position. Die Position ist ja aber falsch wenn man die Liste nicht nach ID sortiert sondern nach Namen oder anderem.

Handy Crash: null object reference

er wurde keine Activity gefunden.

Exception:

FATAL EXCEPTION: Thread-28
com.isabellnoack.myapp, PID: 19749

Process:

```
java.lang.NullPointerException: Attempt to
invoke virtual method 'void
androidx.fragment.app.FragmentActivity.runOnUiThread(java.lang.Runnable)'
on a null object
reference

at
com.isabellnoack.myapp.PokemonFragment.lambda$loadPokemon$3$com-isabellnoack-myapp-
PokemonFragment(PokemonFragment.java:75)

at
com.isabellnoack.myapp.PokemonFragment$$ExternalSyntheticLambda3.run(Unknown Source:2)

at
java.lang.Thread.run(Thread.java:929)
```

The screenshot shows the Android Studio interface. The code editor displays `PokemonFragment.java` with the following code:

```

    71     //Neuer Thread da Hauptthread nicht blockiert werden darf
    72     new Thread(() -> {
    73         try {
    74             Pokemon pokemon = new PokeAPI().requestPokemon(pokemonId); //Funktion der neuen Instanz pokeAPI aufrufen
    75             getActivity().runOnUiThread() -> {
    76                 //UI
    77                 String name = pokemon.name;
    78                 name = character.toUpperCase(name.charAt(0)) + name.substring(1); //Erster Charakter groß geschrieben :
    79                 binding.pokemonName.setText("Name: " + name);
    80                 binding.pokemonExperience.setText("Experience: " + pokemon.baseExperience);
    81                 binding.pokemonId.setText("ID: " + pokemonId);
    82             }
    83         }
    84     });

```

The Logcat tab shows the following log entries:

```

2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    at java.net.URL.<init>(URL.java:498)
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    at java.net.URL.<init>(URL.java:447)
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    at com.isabelnoack.myapp.api.PokeAPI$ImageLoader.loadImageFromUrl(PokeAPI.java:40)
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    at com.isabelnoack.myapp.PokemonFragment$1.run(PokemonFragment.java:97) <1 internal line>
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    at java.lang.Thread.run(Thread.java:929)
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    V querySoundEffectsEnabled...
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    E No adapter attached; skipping layout
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    I [PatchStore] createDisableExceptionQarthFile
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    I [PatchStore] create disable file for com.isabelnoack.myapp uid is 10719
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp    E FATAL EXCEPTION: Thread-28
2024-01-05 11:11:07.077 19749-20226 System.err    com.isabelnoack.myapp, PID: 19749
2024-01-05 11:11:07.077 19749-20226 System.err    java.lang.NullPointerException: Attempt to invoke virtual method 'void androidx.fragment.app.FragmentActivity.runOnUiThread(java.lang.Runnable)' on a null object reference
2024-01-05 11:11:07.077 19749-20226 System.err    at com.isabelnoack.myapp.PokemonFragment.lambda$loadPokemon$3$com-isabelnoack-myapp-PokemonFragment(PokemonFragment.java:75)
2024-01-05 11:11:07.077 19749-20226 System.err    at com.isabelnoack.myapp.PokemonFragment$$ExternalSyntheticLambda3.run(Unknown Source:2) <1 internal line>
2024-01-05 11:11:07.077 19749-20226 System.err    I Sending signal. PID: 19749 SIG: 9
2024-01-05 11:11:07.077 19749-20226 System.err    E notifyActivityState pkg:com.isabelnoack.myapp/com.isabelnoack.myapp.MainActivity state:19 fg:false mUid:10719

```

The Running Devices panel shows a connected Huawei device.

Lösung anscheinend, statt `getActivity()` im Thread aufzurufen, zuvor definieren

This happened when you call `getActivity()` in another thread that finished after the fragment has been removed. The typical case is calling `getActivity()` (ex. for a `Toast`) when an HTTP request finished (in `onResponse` for example).

93

To avoid this, you can define a field name `mActivity` and use it instead of `getActivity()`. This field can be initialized in `onAttach()` method of Fragment as following:

```

@Override
public void onAttach(Context context) {
    super.onAttach(context);

    if (context instanceof Activity){
        mActivity =(Activity) context;
    }
}

```

In my projects, I usually define a base class for all of my Fragments with this feature:

```

public abstract class BaseFragment extends Fragment {

    protected FragmentActivity mActivity;

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
    }
}

```

Kleine Feinheiten von der UI/UX/Navigation:

1. Wenn ich im PokemonFragment die Pokemon durchklicke, soll beim nächsten Mal genau das Pokemon wieder sichtbar sein, wenn man im ListFragment auf den Button Pokemon drückt:

dafür beim Button Next und Previous die Variable „pokemonIdToOpen“ auch updaten.

The screenshot shows the Android Studio project structure on the left and the corresponding Java code on the right. The code is part of the `PokemonFragment` class and handles button click listeners for navigating between Pokemons.

```
//Button NEXT POKEMON
binding.nextPokemonButton.setOnClickListener((view1) -> {
    // laden des nächsten Pokemon
    if (pokemonId == 1025) { //1017 ist das letzte Pokemon
        pokemonId = 1;
        pokemonIdToOpen = 1;
    } else {
        pokemonId++;
        pokemonIdToOpen++;
    }
    loadPokemon();
});

//Button PREVIOUS POKEMON
binding.previousPokemonButton.setOnClickListener((view1) -> {
    // laden des vorherigen Pokemon
    if (pokemonId == 1) {
        pokemonId = 1025;
        pokemonIdToOpen = 1025;
    } else {
        pokemonId--;
        pokemonIdToOpen--;
    }
    loadPokemon();
});

// Pokemon anhand ID wird geladen
loadPokemon();
```

2. Wenn man vom PokemonFragment zum ListFragment wechselt, soll die RecyclerView zum richtigen Item scrollen

Fehlermeldung:

Cannot scroll to position a LayoutManager set. Call setLayoutManager with a non-null argument.

=> Fehler konnte auf die schnelle nicht gelöst werden, daher erstmal weg gelassen

PokemonFragment Design

- Aussehen (Farben)

- Inhalte

- Aufbau

In Figma Design erstellt

Weitere Infos PokemonFragment – Weight, Height, Types

=> Ziel Weight, Height und die Types (Pokemon Art)

=> Pokemon Klasse ergänzt

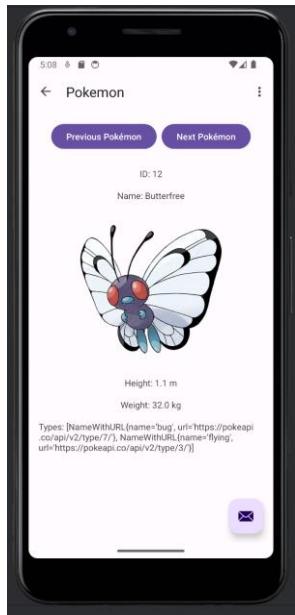
```
public int height = 0;
public int weight = 0;
public ArrayList<NameWithURL> types = new ArrayList<>();
```

=> in der PokeAPI mit abgefragt

```
case "weight":  
    if (reader.peek() == JsonToken.NUMBER) {  
        pokemon.weight = reader.nextInt();  
    } else reader.skipValue();  
    break;  
  
case "height":  
    if (reader.peek() == JsonToken.NUMBER) {  
        pokemon.height = reader.nextInt();  
    } else reader.skipValue();  
    break;  
  
case "types":  
    reader.beginArray();  
    while (reader.hasNext()) {  
        reader.beginObject();  
        while (reader.hasNext()) {  
            switch (reader.nextName()) {  
                case "type":  
                    pokemon.types.add(readNameWithURL(reader));  
                    break;  
                default:  
                    reader.skipValue();  
            }  
        }  
        reader.endObject();  
    }  
    reader.endArray();  
    break;
```

=> in PokemonFragment Binding erstellt

```
if (pokemon.weight != 0) {  
    binding.pokemonWeight.setText("Weight: " + (pokemon.weight / 10.0f) + " kg"); //Angabe in  
    Hectogramm // ausserdem mit float multiliziert, für Kommastellen  
} else {  
    binding.pokemonWeight.setText("Weight: undefined");  
}  
  
if (pokemon.height != 0) {  
    binding.pokemonHeight.setText("Height: " + (pokemon.height / 10.0f) + " m"); //Angabe in Decimeter  
} else {  
    binding.pokemonHeight.setText("Weight: undefined");  
}  
  
binding.pokemonTypes.setText("Types: " + pokemon.types);
```



Problem: Array wird noch nicht richtig angezeigt (soll nur die Namen zeigen, nicht alle Informationen aus dem Array)

=> for-each Schleife, um den Namen in einen anderen String zu speichern

```
//pokemon.types, nur name aus dem array ausgeben
String types = "none";
for (NameWithURL type : pokemon.types) {
    //for (Typ variablen-name : Datenquelle)
    if (types.equals("none")) {
        types = capitalize(type.name);                                //Wert überschreiben mit Pokemon Type
    } else {
        types = types + ", " + capitalize(type.name);      //Pokemon Type anhängen
    }
}
binding.pokemonTypes.setText("Types: " + types);
```

IO Exception behandeln

IO Exception wenn man kein Internet hat oder dieses abbricht, oder Server ausfällt, File not found, etc.

```
} catch (IOException ignored) { //Hier im Download Thread, um etwas anzeigen zu können in den UI Thread wechseln
    // todo send warning to user
}
```

Hier im Download Thread, um etwas anzeigen zu können in den UI Thread wechseln

```
activity.runOnUiThread(() -> { //jetzt auf UI Thread
    // todo send warning to user
});
```

Jetzt Fehlermeldung anzeigen:

Klasse “Toast” benutzen – Fehlermeldung als kleiner Kasten der aufploppt

Das Toast zeigt die passende Exception für uns an!!

– Das macht uns aus dem Fehlertext lesbare UI

```
} catch (IOException exception) { //Hier im Download Thread, um etwas anzeigen zu können in den UI Thread wechseln
    activity.runOnUiThread(() -> { //jetzt auf UI Thread
        // send warning to user
        Toast.makeText(activity, exception.toString(),
        Toast.LENGTH_LONG).show(); //Toast Klasse mit: context(so anzeigen: activity ; Fehler als String anzeigen; Wie lange angezeigt)
    });
}
```

Der Fehlertext kommt aus exception.toString()

Weitere Infos PokemonFragment - Ability

Neue Klasse Ability

```
public ArrayList<Ability> abilities = new ArrayList<>();
```

```
public class Ability {
    public NameWithURL nameWithURL;
    public boolean isHidden;

    @Override
    public String toString() {
        return "Ability{" +
            "nameWithURL=" + nameWithURL +
            ", isHidden=" + isHidden +
            '}';
    }
}
```

=> Datenstruktur fertig definiert

In PokeAPI, Werte in Datenstruktur füllen

```
case "abilities":
    reader.beginArray();
    while (reader.hasNext()) {
        pokemon.abilities.add(readAbility(reader));
    }
    reader.endArray();
    break;
```

neue Funktion damit der Text übersichtlicher ist

```
Ability readAbility(JsonReader reader) throws IOException { //Rückgabewert
Ability
    Ability ability = new Ability();
    reader.beginObject();
    while (reader.hasNext()) {
        switch (reader.nextName()) {
```

```

        case "ability":
            ability.nameWithURL = readNameWithURL(reader);
            break;
        case "is_hidden":
            ability.isHidden = reader.nextBoolean();
            break;
        default:
            reader.skipValue();
    }
}
reader.endObject();
return ability;
}

```

=> Parsing fertig

Jetzt Darstellung (UI)

Layout und PokemonFragment Binding

Binding:

```

//pokemon.abilities, nur name aus dem array ausgeben, und nach hidden filtern
String abilities = "none";
int numberofAbilities = 0;
for (Ability ability : pokemon.abilities) {
    NameWithURL nameWithURL = ability.nameWithURL;
    if (ability.isHidden) {
        continue;
    }
    //macht mit dem for weiter, und ignoriert damit die Ability wenn isHidden true
    numberofAbilities++;
    if (abilities.equals("none")) {
        abilities = capitalize(nameWithURL.name);
    } else {
        abilities = abilities + ", " + capitalize(nameWithURL.name);
    }
}
if (numberofAbilities == 1) {
    binding.pokemonAbilities.setText("Ability: " + abilities);
} else {
    binding.pokemonAbilities.setText("Abilities: " + abilities);
}

```

Farbanpassungen Buttons

In Ordner values – colors definieren (colors.xml)

In Unterordner themes – colors zuweisen (themes.xml und themes.xml night version anpassen)

Schütteln-Sensor

Um schütteln zu erkennen: Accelerometer

SensorManager integriert mithilfe <https://www.youtube.com/watch?v=gVsXHio7hU> und PDF

Code – Button ausführen

```
•     public MainActivityFragment() {      }
•     @Override
•     public View onCreateView(LayoutInflater inflater, ViewGroup container,
•         Bundle savedInstanceState) {
•         mSensorManager = (SensorManager) getActivity().getSystemService(Context.SENSOR_SERVICE);
•         mAccelerometer = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
•
•         View view = inflater.inflate(R.layout.fragment_main, container, false);
•         Button button = (Button) view.findViewById(R.id.btnGPSlocation);
•         button.setOnClickListener(new View.OnClickListener() {
•             @Override
•             public void onClick(View v) {
•                 // do something
•             }
•         });
•         return view;
•     }
```

Prof. Dr.-Ing. Michael Stepping, Cloud and Mobile Computing

157

www.eah-jena.de

Sensor Manager

```
SensorManager sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
Sensor sensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
sensorManager.registerListener(this, sensor, SensorManager.SENSOR_DELAY_NORMAL);
```

Um den Code ein try und catch um eine Fehlermeldung zu behandeln: falls das Handy keinen Sensor oder SensorManager hat (NullPointerException) dann soll dieses nicht abstürzen sondern einfach nichts machen (ignored)

Im Pokemon Fragment noch implements SensorEventListener ergänzt

```
public class PokemonFragment extends Fragment implements SensorEventListener {
```

Schütteln erkennen

Erklärung in den Kommentaren

```
//Sensor
private static final float SHAKE_THRESHOLD = 30.0f; //g Beschleunigung
private long lastShakeTime;

@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        binding.sensor.setText("X: " + event.values[0] + ", Y: " +
event.values[1] + ", Z: " + event.values[2]);
    }

    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        // Beschleunigungssensors Werte
        float x = event.values[0];
        float y = event.values[1];
        float z = event.values[2];

        // Absolute Beschleunigung berechnen (Betrag der Differenz zur Erd-
```

```

beschleunigung)
    // .abs = absoluter Betrag: heißt es geht in + oder -
    float acceleration = Math.abs(x) + Math.abs(y) + Math.abs(z) -
(SensorManager.GRAVITY_EARTH); //Gravitation 9,81 m/s2, auf Z wenn es flach
auf Tisch liegt

    // Die aktuelle Zeit in Millisekunden
    long currentTime = System.currentTimeMillis();

    // Überprüfe, ob die absolute Beschleunigung den Schwellenwert für
ein Schütteln überschreitet
    if (acceleration > SHAKE_THRESHOLD) {

        // Cool-Down: Überprüfe, ob seit dem letzten Schütteln eine be-
stimmte Zeit vergangen ist
        if (currentTime > lastShakeTime + 1000) {           //Momentane
Zeit ist größer als letzteShake Zeit + 1sek
            onShakeDetected();                                //Wenn ja, wird
Methode aufgerufen
            lastShakeTime = currentTime;
        }
    }
}
}

```

Wenn schütteln erkannt wurde, dann: (Toast zum Testen)

```

//Schütteln erkannt
private void onShakeDetected() {
    Toast.makeText(getApplicationContext(), "Shake Detected!",
Toast.LENGTH_SHORT).show();
}

```

Pokemon schütteln

Den Toast jetzt dazu ändern, dass sich stattdessen das Pokemon Bild bewegt 😊

=> Damit sich das Pokemon Bild schüttelt: ImageView soll animiert werden

360 Rotation:

```

ObjectAnimator rotationAnimator = ObjectAnimator-
ofFloat(binding.pokemonImage, "rotation", 0f, 720f);
rotationAnimator.setDuration(1000); // Dauer der Rotation in Millisekunden
(hier 500 Millisekunden)
rotationAnimator.start();

```

Stattdessen hin und her wackeln, ObjectAnimation Zeile austauschen:

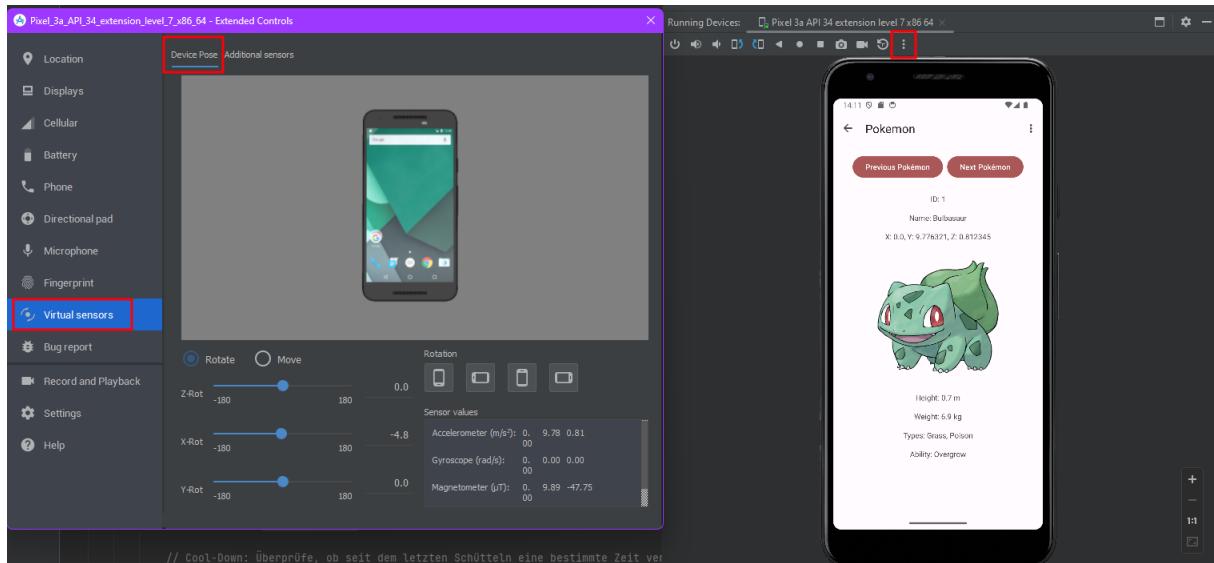
```

ObjectAnimator rotationAnimator = ObjectAnimator.ofFloat(
binding.pokemonImage, "rotation", 0f, -20f, 20f, -20f, 0f);

```

Simulierte Handy testen

Schütteln des Handys testen über: (Eventuell SHAKE_THRESHOLD anpassen)



Pokemon Species: Neue URL einbauen

<https://pokeapi.co/api/v2/pokemon-species/1/>

Das will ich abfragen, dafür kann ich vieles von der Pokemon Abfrage für die PokemonSpecies abfrage benutzen! – d.h. alles kopiert und entleert

- In PokeAPI
 - o public PokemonSpecies requestPokemonSpecies(int id)
 - o PokemonSpecies readPokemonSpecies(JsonReader reader)
- Neue Klasse public class PokemonSpecies
- Im PokemonFragment neues void loadPokemonSpecies(); erstellt und aufgerufen in parallel laufendem Thread
 - o Passende Bindings

=> jetzt die passenden Werte abfragen

```
public class PokemonSpecies {
    public PokemonSpecies() {
    }

    public String name = "Unknown";

    public ArrayList<FlavorTextEntry> flavorTextEntries = new Ar-
    rayList<>();
}
```

```
public class FlavorTextEntry {
    public String flavorText;
    public NameWithURL language;
    public NameWithURL version;

    @Override
    public String toString() {
        return "FlavorTextEntry{" +
            "flavorText='" + flavorText + '\'' +
            ", language='" + language +
            ", version='" + version +'
```

```

        '}' ;
    }
}

PokemonSpecies readPokemonSpecies(JsonReader reader) throws IOException {
    PokemonSpecies pokemonSpecies = new PokemonSpecies();
    reader.beginObject();
    while (reader.hasNext()) {
        switch (reader.nextName()) {
            case "name":
                pokemonSpecies.name = reader.nextString(); //Formatierung:
Strg+Alt+O Strg+Alt+L
                break;

            case "flavor_text_entries":
                reader.beginArray();
                while (reader.hasNext()) {
                    pokemonSpe-
cies.flavorTextEntries.add(readFlavorTextEntry(reader));
                }
                reader.endArray();
                break;

            default:
                reader.skipValue();
        }
    }
    reader.endObject();
    return pokemonSpecies;
}

```

```

private FlavorTextEntry readFlavorTextEntry(JsonReader reader) throws IOException {
    FlavorTextEntry flavorTextEntry = new FlavorTextEntry();
    reader.beginObject();
    while (reader.hasNext()) {
        switch (reader.nextName()) {
            case "flavor_text":
                flavorTextEntry.flavorText = reader.nextString();
                break;
            case "language":
                flavorTextEntry.language = readNameWithURL(reader);
                break;
            case "version":
                flavorTextEntry.version = readNameWithURL(reader);
                break;
            default:
                reader.skipValue();
        }
    }
    reader.endObject();
    return flavorTextEntry;
}

```

=> Danach Binding, Text ausgeben anhand language und version

=> Per Klick auf den Text eine andere Textversion ausgeben, aber auch dazu schreiben welche Version das ist

PokemonSpecies: FlavorTextEntry (Array List)

Array List da es im JSON eckige Klammern statt geschwungene hat: [...]

In PokemonSpecies:

```
public ArrayList<FlavorTextEntry> flavorTextEntries = new ArrayList<>();
```

Neue Klasse

```
public class FlavorTextEntry {
    public String flavorText;
    public NameWithURL language;
    public NameWithURL version;

    @Override
    public String toString() {
        return "FlavorTextEntry{" +
            "flavorText='" + flavorText + '\'' +
            ", language=" + language +
            ", version=" + version +
            '}';
    }
}
```

In der API:

Wir lesen nur die englischen Texte und speichern sie in der Array List als Typ flavorTextEntry

```
case "flavor_text_entries":
    reader.beginArray();
    while (reader.hasNext()) {
        FlavorTextEntry flavorTextEntry = readFlavorTextEntry(reader);
        if (flavorTextEntry.language.name.equals("en")) { // nur
englische Texte speichern
            pokemonSpecies.flavorTextEntries.add(flavorTextEntry);
        }
    }
    reader.endArray();
    break;
```

Ausgabe im PokemonFragment

Hier gehen wir das Array durch

Mit dem OnClickListener (wenn jemand klickt) erhöht sich der Index und mit dem neuen Index wird der nächste Text in unseren String „flavorText“ geschrieben und ausgegeben

```
// PokemonSpecies.flavorTextEntries
versionIndex = 0;
String flavorText = "none";
String version = "";
if (!pokemonSpecies.flavorTextEntries.isEmpty()) { //Liste nicht empty,
dann:
    FlavorTextEntry flavorTextEntry = pokemonSpecies.flavorTextEntries.get(0); //Erster Eintrag
```

```

        flavorText = flavorTextEntry.flavorText;
        version = flavorTextEntry.version.name;
    }
binding.pokemonSpeciesFlavorText.setText("'" + flavorText.replace("\n", " "
) +
        "'" + "\n Version " + version);
binding.pokemonSpeciesFlavorText.setOnClickListener(v -> {
    if (++versionIndex == pokemonSpecies.flavorTextEntries.size()) {
        versionIndex = 0;
    }
    FlavorTextEntry flavorTextEntry = pokemonSpecies-
    .flavorTextEntries.get(versionIndex);
    String flavorText1 = flavorTextEntry.flavorText;
    binding.pokemonSpeciesFlavorText.setText("'" + flavor-
    Text1.replace("\n", " ") +
        "'" + "\n Version " + flavorTextEntry.version.name);
});

```

Schriftart

Neuer Font Ordner

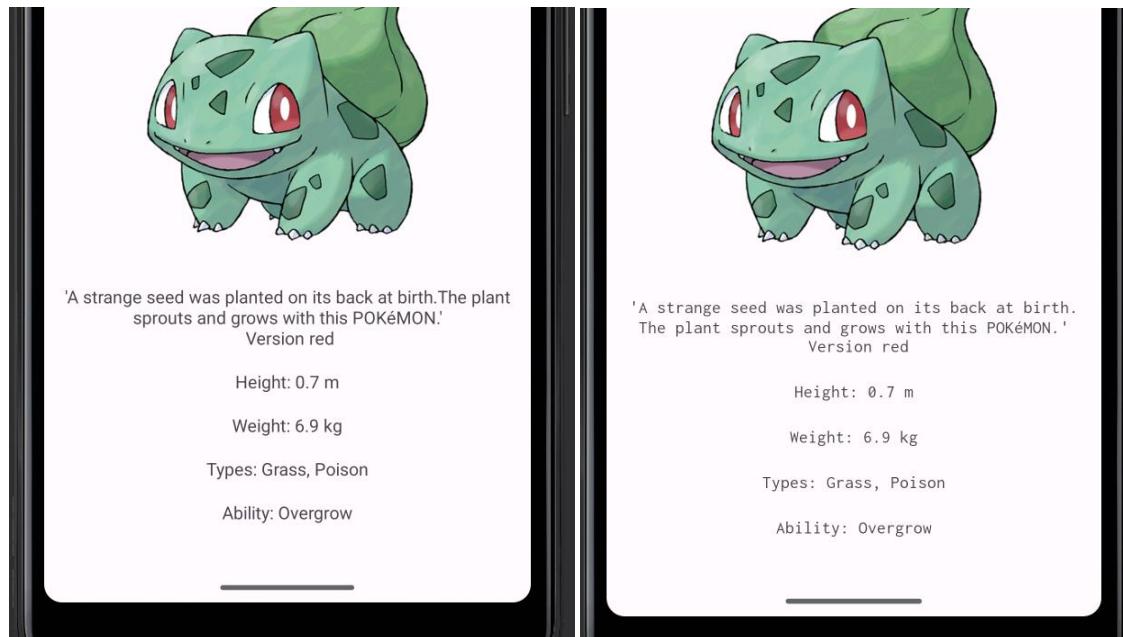
Darein die .ttf

!Achtung bei der Benennung, nur Kleingeschrieben und kein Minus

In Datei themes.xml

```
<item name="android:fontFamily">@font/inconsolataregular</item>
```

Font: **Inconsolata** (Google Font designed by Raph Levien)



Größenänderung:

In den dimens.xml

```
<dimen name="button_size">18sp</dimen>
<dimen name="text_size_regular">12sp</dimen>
```

In der layout.xml dann diese Werte zuweisen – Beispiel Button:

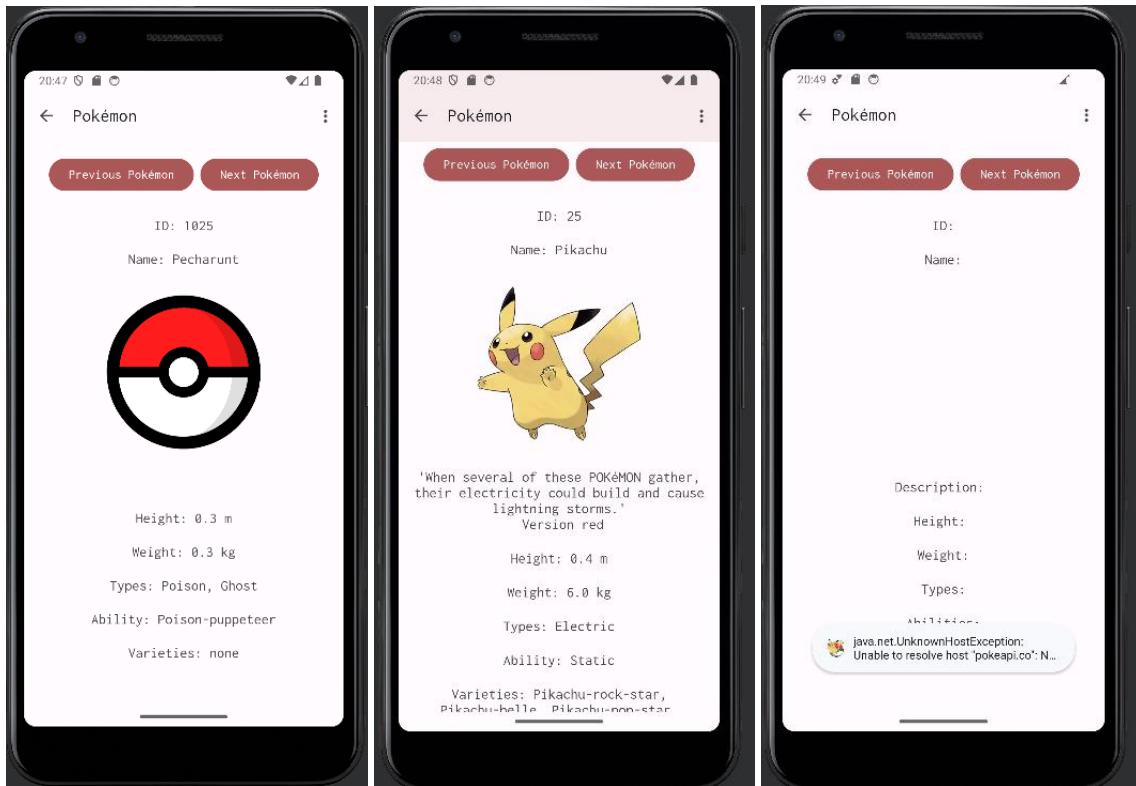
```
android:textSize="@dimen/button_size"
```



Stand am Ende

The image displays three screenshots of a mobile application for Pokédex-like data. The first two screenshots show a list of the first six Pokémon: Bulbasaur, Ivysaur, Venusaur, Charmander, Charmeleon, and Charizard. Each entry includes the name, ID, and a small icon. The third screenshot shows the detailed view for Ivysaur (ID 2), featuring:

- A blue, leafy Pokémon with a large bulb on its back.
- ID: 2
- Name: Ivysaur
- 'When the bulb on its back grows large, it appears to lose the ability to stand on its hind legs.'
- Version red
- Height: 1.0 m
- Weight: 13.0 kg
- Types: Grass, Poison
- Ability: Overgrow
- Varieties: none



Ausblick

Wenn man das Projekt fortsetzen würde, wären das Ideen zum fortsetzen:

- weitere Daten: wie Entwicklung der Pokemon
- Von den Variationen die vorhandenen Bilder in einer Liste
- Caching für besseres Laden
- Suche und Filtern nach Name, ID, Eigenschaften
- Pokemon Favoriten speichern
- verschiedene Sprachen unterstützen
- Design überarbeiten

Weitere Datein

Vortrag Skript: <https://md.gafert.org/s/MUPoZI1ZK>

Übersicht des Codes (Modell) mit draw.io erstellt (Modell.drawio / .pdf)