

IPW

UAIM

Laboratorium 6

Karol Żelazowski 324953

Spis treści

1. Opis kodu	3
1.1. Aktywności	3
1.1.1. Klasa MainActivity	3
1.1.2. Klasa DetailsActivity	6
1.2. Pliki XML	8
1.2.1. activity_main	8
1.2.2. activity_details	8
1.2.3. item_artwork	9
2. Funkcjonalność	11
2.1. Widok ogólny	11
2.2. Widok szczegółowy	11
2.3. Toast	13

1. Opis kodu

1.1. Aktywności

1.1.1. Klasa MainActivity

```
package com.example.lab_6;

import android.os.Bundle;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    private ArrayList<Artwork> artworks = new ArrayList<>();
    private ArtworkAdapter adapter;
    private static final String TAG = "MainActivity"; // Tag dla logów

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

Java

```

        ListView listView = findViewById(R.id.listViewArtworks);
        adapter = new ArtworkAdapter(this, artworks);
        listView.setAdapter(adapter);

        fetchArtData();

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
                Artwork artwork = artworks.get(position);
                Intent intent = new Intent(MainActivity.this,
DetailsActivity.class);
                intent.putExtra("title", artwork.getTitle());
                intent.putExtra("imageBase64", artwork.getImageBase64());
                intent.putExtra("description", artwork.getDescription());
                startActivity(intent);
            }
        });
    }

    private void fetchArtData() {
        String url = "https://api.artic.edu/api/v1/artworks?page=2&limit=10";
        Log.d(TAG, "Making API request to: " + url);
        RequestQueue queue = Volley.newRequestQueue(this);

        JsonObjectRequest request = new JsonObjectRequest(Request.Method.GET, url,
null,
            new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {
                    Log.d(TAG, "API Response received: " + response.toString());
                    try {
                        // Parse the "data" part of the response to get the
artwork details
                        JSONArray artworksArray =
response.getJSONArray("data"); // Correctly extract the "data" array
                        for (int i = 0; i < artworksArray.length(); i++) {
                            JSONObject artworkJson =
artworksArray.getJSONObject(i);
                            String title = artworkJson.getString("title");
                            String description =
artworkJson.optString("description", "Brak opisu");
                            JSONObject thumbnailJson =
artworkJson.optJSONObject("thumbnail");
                            String base64Image = null;

```

```

        if (thumbnailJson != null) {
            base64Image = thumbnailJson.optString("lqip");
        }

        // If image exists, add artwork to the list
        if (base64Image != null && !base64Image.isEmpty()) {
            artworks.add(new Artwork(title, base64Image,
description));

            Log.d(TAG, "Artwork added: " + title);
        } else {
            Log.w(TAG, "No image found for artwork: " +
title);
        }
    }
    adapter.notifyDataSetChanged();
    Log.d(TAG, "Adapter notified, UI updated");
} catch (JSONException e) {
    e.printStackTrace();
    Log.e(TAG, "Error while processing response: " +
e.getMessage());
}
},
new Response.ErrorListener() {
    @Override
    public void onResponse(VolleyError error) {
        error.printStackTrace();
        Log.e(TAG, "Error fetching data: " + error.getMessage());
    }
});

queue.add(request);
}
}

```

MainActivity jest punktem startowym aplikacji. Jej głównym zadaniem jest wyświetlenie listy dzieł sztuki pobranych z API i umożliwienie użytkownikowi przejścia do szczegółów wybranego dzieła.

Główne elementy:

- Lista dzieł sztuki (ListView):
- Zidentyfikowana jako listViewArtworks w pliku XML.
- Wyświetla listę dzieł sztuki za pomocą niestandardowego adaptera ArtworkAdapter.
- Obsługuje kliknięcia elementów listy.

Szczegóły implementacji:

- Inicjalizacja ListView:
 - Po podłączeniu układu XML (setContentView(R.layout.activity_main)), inicjalizowany jest ListView za pomocą identyfikatora R.id.listViewArtworks.
 - Tworzony jest adapter ArtworkAdapter, który wykorzystuje niestandardowy układ dla każdej pozycji listy (np. list_item.xml).
- Pobieranie danych z API (fetchArtData):
 - URL API: <https://api.artic.edu/api/v1/artworks?page=2&limit=10>.
 - Wykorzystanie biblioteki Volley do wykonania żądania typu GET.
- Proces przetwarzania odpowiedzi:
 - Odpowiedź JSON zawiera klucz data, który jest tablicą dzieł sztuki.
 - Z każdego dzieła wyciągane są dane: tytuł (title), opis (description) oraz obraz w formacie Base64 (thumbnail.lqip).
 - Jeżeli obraz jest dostępny, tworzony jest obiekt Artwork i dodawany do listy artworks.
- Obsługa kliknięć na elementy listy:
 - Każde kliknięcie elementu listy wywołuje przejście do DetailsActivity.
 - Wysłane dane to:
 - title: Tytuł dzieła sztuki.
 - imageBase64: Obraz w formacie Base64.
 - description: Opis dzieła.

Logowanie i diagnostyka: Każde ważne zdarzenie, takie jak żądanie API, przetwarzanie odpowiedzi czy błędy, jest logowane z użyciem Log:

- Log.d(TAG, „opis zdarzenia”) – dla informacji diagnostycznych.
- Log.e(TAG, „opis błędu”) – dla błędów.

1.1.2. Klasa DetailsActivity

```
package com.example.lab_6;

import android.os.Bundle;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.util.Base64;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.widget.Toast;

public class DetailsActivity extends AppCompatActivity {
```

Java

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_details);

    String title = getIntent().getStringExtra("title");
    String base64Image = getIntent().getStringExtra("imageBase64");
    String description = getIntent().getStringExtra("description");

    TextView titleView = findViewById(R.id.textViewDetail);
    ImageView imageView = findViewById(R.id.imageViewDetail);
    TextView descriptionView = findViewById(R.id.artworkDescription);

    titleView.setText(title);
    descriptionView.setText(description);

    View backButton = findViewById(R.id.backButton);

    backButton.setOnClickListener(v -> {
        // Wysyłanie wiadomości Toast z treścią obrazu po powrocie
        Toast.makeText(DetailsActivity.this, "Wrócono do listy: " + title,
            Toast.LENGTH_SHORT).show();
        finish(); // Powrót do poprzedniej aktywności
    });

    byte[] decodedString = Base64.decode(base64Image.split(",")[1],
        Base64.DEFAULT);
    Bitmap decodedBitmap = BitmapFactory.decodeByteArray(decodedString, 0,
        decodedString.length);
    imageView.setImageBitmap(decodedBitmap);
}
}

```

DetailsActivity wyświetla szczegółowe informacje o wybranym dziele sztuki: obraz, tytuł oraz opis.

Główne elementy:

- Obraz dzieła (ImageView): Wyświetla obraz dzieła sztuki przetworzony z Base64 na format bitmapy.
- Tytuł dzieła (TextView): Wyświetla tytuł dzieła.
- Opis dzieła (TextView): Wyświetla szczegółowy opis dzieła.
- Przycisk powrotu (Button): Umożliwia powrót do MainActivity i wyświetla komunikat Toast z informacją o nazwie dzieła.

Szczegóły implementacji:

- Pobieranie danych z Intent:
 - DetailsActivity odbiera dane przekazane przez MainActivity za pomocą Intent.
 - Dane są przypisane do zmiennych:

- title (tytuł),
- base64Image (obraz w formacie Base64),
- description (opis).
- Wyświetlanie danych:
 - Tytuł i opis: Przypisywane do odpowiednich widoków TextView (textViewDetail i artworkDescription).
 - Obraz:
 - Obraz w formacie Base64 jest dekodowany za pomocą Base64.decode().
 - Wynik jest przekształcany na bitmapę za pomocą BitmapFactory.decodeByteArray() i ustawiany w ImageView.
- Przycisk powrotu:
 - Zidentyfikowany jako backButton.
 - Obsługuje kliknięcia, po których:
 - Wyświetlany jest Toast z informacją o powrocie i tytule dzieła.
 - Aktywność jest zamykana (finish()), co powoduje powrót do MainActivity.

1.2. Pliki XML

1.2.1. activity_main

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <ListView
        android:id="@+id/listViewArtworks"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:dividerHeight="1dp"
        android:divider="@android:color/darker_gray"
    />

</LinearLayout>
```

XML

Zawiera tylko ListView (listViewArtworks) do wyświetlania listy dzieł sztuki.

1.2.2. activity_details

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <ImageView
        android:id="@+id/imageViewDetail"
        android:layout_width="match_parent"
```

XML


```

        android:layout_height="300dp"
        android:layout_centerHorizontal="true" />

<TextView
    android:id="@+id/textViewDetail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/imageViewDetail"
    android:textSize="18sp"
    android:padding="16dp" />

<TextView
    android:id="@+id/artworkDescription"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp" />

<Button
    android:id="@+id/backButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Back"
    android:layout_marginTop="16dp"
    android:layout_gravity="center" />

</LinearLayout>

```

Wyświetla szczegóły wybranego dzieła sztuki.

Elementy:

- ImageView (imageViewDetail) – duży obraz dzieła.
- TextView (textViewDetail) – tytuł dzieła.
- TextView (artworkDescription) – opis dzieła.
- Button (backButton) – przycisk „Back” do powrotu do listy.

1.2.3. item_artwork

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="16dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:layout_alignParentLeft="true"

```

XML

```
        android:layout_centerVertical="true"
        android:src="@drawable/ic_launcher_foreground" />

<TextView
    android:id="@+id/textViewTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:layout_toEndOf="@id/imageView"
    android:paddingLeft="16dp"
    android:text="Title" />
</LinearLayout>
```

Reprezentuje pojedyncze dzieło sztuki w ListView.

Elementy:

- ImageView (imageView) – miniatura dzieła sztuki.
- TextView (textViewTitle) – tytuł dzieła.

2. Funkcjonalność

2.1. Widok ogólny



Rysunek 1: Ogólny widok listy z dziełami sztuki oraz tytułami

2.2. Widok szczegółowy

W widoku szczegółowy wyświetlany jest obraz oraz jego tytuł wraz z opisem dzieła. Dostępny jest również przycisk „Back”, który cofa nas do głównej aktywności



Rysunek 2: Detale dotyczące dzieła sztuki

2.3. Toast



Rysunek 3: Komunikat Toast po wyjściu z szczegółowego opisu