

# Aplikacja mobilna jako natywna aplikacja dla systemu Android - komunikacja sieciowa

Z wykorzystaniem Android Studio  
ARIUS Zadanie 6.

**Julia Chilczuk 325261**

Politechnika Warszawska  
Wydział Elektroniki i Technik Informacyjnych

24 maja 2025

## Spis treści

<b>1. Wstęp</b>	2
<b>2. Struktura aplikacji</b>	2
2.1. Aktywności Java	2
2.1.1. MainActivity.java	2
2.1.2. ArtDetailsActivity.java	2
2.2. Klasy pomocnicze	2
2.2.1. Artwork.java	2
2.2.2. ArtworkAdapter.java	2
2.3. Układy widoków XML	2
2.3.1. activity_main.xml	2
2.3.2. single_row.xml	3
2.3.3. activity_art_details.xml	3
<b>3. Efekty działania aplikacji</b>	3
3.1. Wyświetlanie listy obiektów pobranych z API	3
3.2. Wyświetlanie szczegółów obiektu	4
<b>4. Pliki źródłowe</b>	4

## 1. Wstęp

Aplikacja została zbudowana w środowisku Android Studio jako natywna aplikacja w języku Java. Jej zadaniem jest pobieranie danych dotyczących dzieł Vincenta van Gogha z publicznego API The Metropolitan Museum of Art (<https://metmuseum.github.io/>) i ich prezentacja w postaci listy obrazów z możliwością wyświetlenia szczegółów.

## 2. Struktura aplikacji

### 2.1. Aktywności Java

#### 2.1.1. MainActivity.java

Główna aktywność aplikacji odpowiada za pobranie danych z zewnętrznego API i wyświetlenie listy obiektów w komponencie ListView. W tym celu wykorzystano klasę AsyncTask, która w tle wykonuje zapytania HTTP, przetwarza JSON i przekazuje dane do adaptera.

Zapytanie:

```
https://collectionapi.metmuseum.org/public/collection/v1/search?hasImages=true&q=Vincent%20van%20Gogh
```

zwraca listę id obiektów, które zawierają obraz oraz frazę "Vincent van Gogh".

Zapytanie:

```
https://collectionapi.metmuseum.org/public/collection/v1/objects/\[id\]
```

zwraca dane dotyczące obiektu o podanym id. Wczytywanie danych ograniczone jest do 20 obiektów, gdyż w przypadku większej ilości pobieranie trwa proporcjonalnie dłużej.

W widoku głównym w poszczególnych elementach listy wyświetlana jest grafika - kopia obrazu, tytuł, rok powstania i technika malarska.

#### 2.1.2. ArtDetailsActivity.java

Druga aktywność wyświetla szczegóły wybranego obrazu artystycznego: tytuł, autora, technikę malarską, wymiary obrazu oraz rok utworzenia. Informacje są przekazywane poprzez Intent z MainActivity. Do załadowania obrazów wykorzystano bibliotekę Picasso.

Po wciśnięciu przycisku Close następuje powrót do głównej aktywności aplikacji oraz pojawia się komunikat typu „Toast” z treścią odpowiadającą treści wyświetlanej na klikniętym elemencie listy.

### 2.2. Klasy pomocnicze

#### 2.2.1. Artwork.java

Klasa modelowa, reprezentująca dane pojedynczego dzieła sztuki. Zawiera pola: tytuł, URI obrazu, autor, technika, wymiary i rok utworzenia. Obiekt tej klasy jest tworzony dla każdego rekordu API spełniającego warunki (obecność obrazu i tytułu).

#### 2.2.2. ArtworkAdapter.java

Adapter dziedziczy po ArrayAdapter<Artwork> i odpowiada za konwersję danych obiektów do widoku listy. Wewnątrz metody getView() ustawiane są wartości pól oraz obraz za pomocą Picasso.

### 2.3. Układy widoków XML

#### 2.3.1. activity\_main.xml

Zawiera TextView dla tytułu i podtytułu, Progress bar, TextView dla tekstu sygnalizującego postępy pobierania danych oraz główny ListView, który wyświetla listę dzieł sztuki pobranych z API.

### 2.3.2. single\_row.xml

Szablon pojedynczej wiersza w liście, zawiera ImageView dla obrazu, oraz TextView dla tytułu, daty oraz techniki malarskiej.

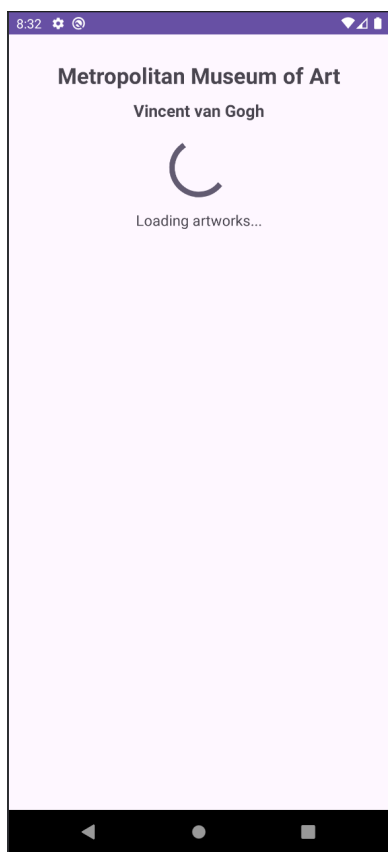
### 2.3.3. activity\_art\_details.xml

Układ dla ArtDetailsActivity, zawiera ImageView oraz TextView dla tytułu, artysty, techniki wymiarów i daty, a także przycisk powrotu (Close).

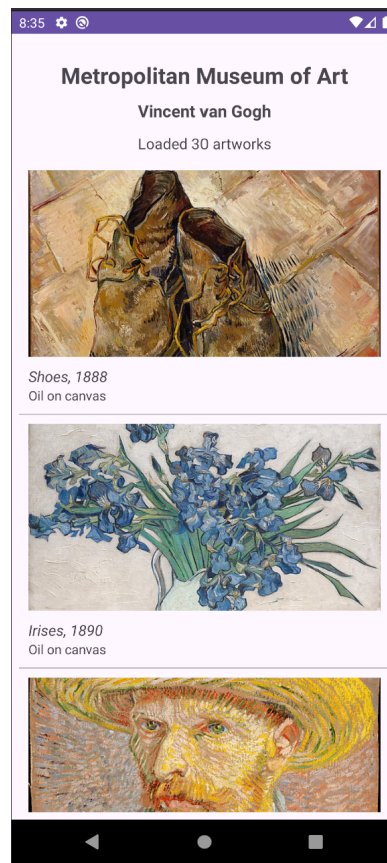
## 3. Efekty działania aplikacji

### 3.1. Wyświetlanie listy obiektów pobranych z API

Poniżej widok podczas procesu pobierania danych z API (po lewej stronie) oraz lista załadowanych obiektów (po prawej).



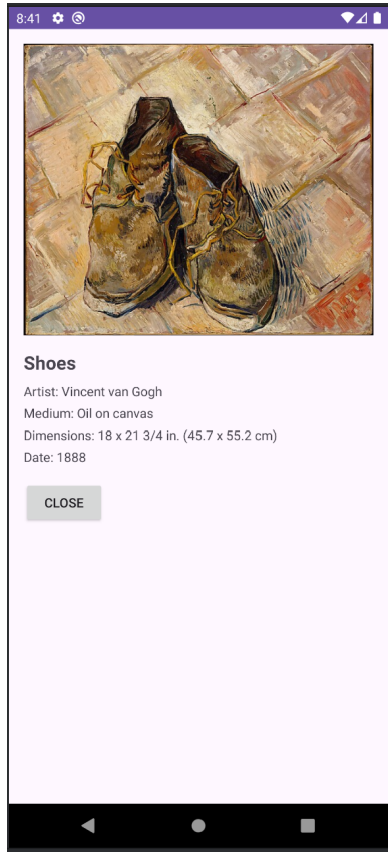
Rys. 1: Widok podczas procesu pobierania danych z API.



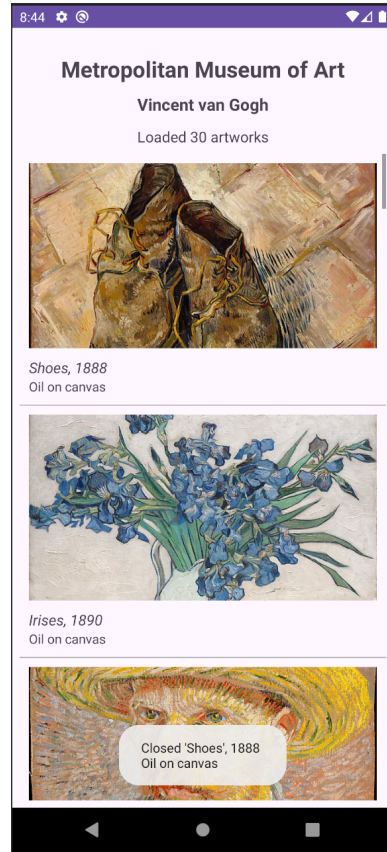
Rys. 2: Widok listy załadowanych obiektów.

### 3.2. Wyświetlanie szczegółów obiektu

Poniżej po lewej stronie widok szczegółowy obiektu o tytule "Shoes", po prawej stronie widok po powrocie do głównej aktywności wyświetlający komunikat typu „Toast” z treścią odpowiadającą treści wyświetlanej na klikniętym elemencie listy.



Rys. 3: Widok szczegółowy obiektu o tytule "Shoes".



Rys. 4: Widok po powrocie do głównej aktywności wyświetlający komunikat typu „Toast”.

### 4. Pliki źródłowe

Do sprawozdania dołączam pliki źródłowe Java oraz XML (znajdujące się w katalogu Main), a także plik *build.gradle.kts* oraz *libs.versions.toml*, który zawiera m.in. deklarację zależności biblioteki Picasso.