

# APLIKACJE MOBILNE

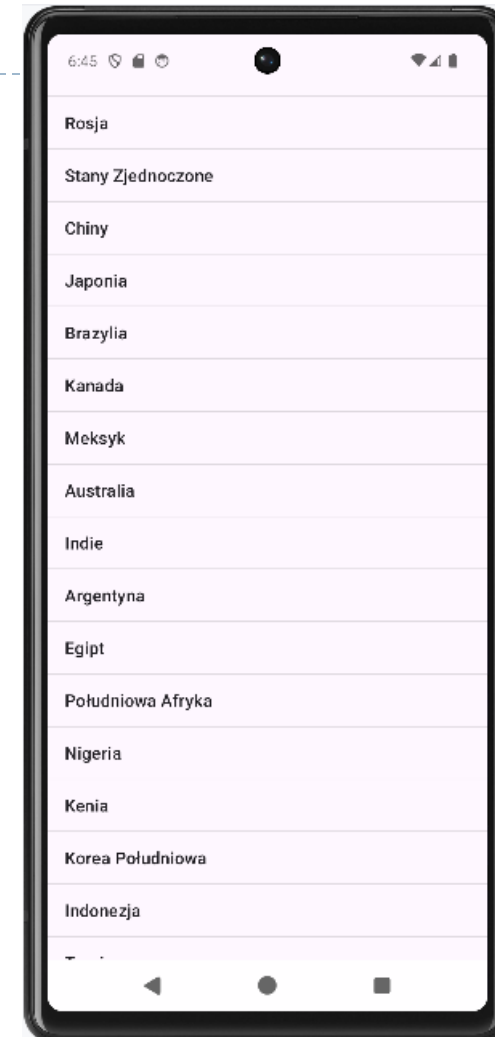
## WIDOK LISTY

dr Artur Bartoszewski

# Widok listy

Komponent **ListView** odpowiada za wyświetlenie dowolnych elementów w postaci listy.

Kiedy elementów jest więcej niż może pomieścić ekran, tworzy się pasek przewijania.



## 1. W pliku XML dodajemy kontrolkę ListView

```
<ListView  
    android:id="@+id/listView01"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
/>
```

## 2. W pliku .java (globalnie)

```
private ListView listView;  
private String[] dane = {  
    "Polska", "Niemcy", "Francja", "Hiszpania", "Włochy", "Ukraina",  
    "Wielka Brytania", "Rosja", "Stany Zjednoczone", "Chiny", "Japonia",  
    "Brazylia", "Kanada", "Meksyk", "Australia", "Indie",  
    "Argentyna", "Egipt", "Południowa Afryka", "Nigeria", "Kenia",  
    "Korea Południowa", "Indonezja", "Turcja", "Arabia Saudyjska", "Iran",  
    "Wietnam", "Tajlandia", "Filipiny", "Pakistan", "Kolumbia"  
};
```

1. Znajdujemy uchwyt do ListView.
2. Przygotowujemy tablicę wartości do wyświetlenia (tablicę można także stworzyć w zasobach).

### 3. W pliku .java (np. w metodzie onCreate)

```
listView = findViewById(R.id.listView01);  
    ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, dane);  
    listView.setAdapter(adapter);
```

Tworzy adapter typu **ArrayAdapter**, który będzie odpowiedzialny za wyświetlanie danych na liście.

- **this** odnosi się do kontekstu aktywności.
- **android.R.layout.simple\_list\_item\_1** to wbudowany layout dla pojedynczego elementu listy (prosty tekst).
- **dane** to tablica z danymi do wyświetlenia.

Polecenia **listView.setAdapter(adapter);** ustawia adapter dla ListView, co spowoduje wyświetlenie danych na liście.

### 3. Reakcja listy na kliknięcie w element

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
        Toast.makeText(MainActivity.this, "Pozycja:" + position + " id:" + id,  
            Toast.LENGTH_SHORT).show();  
    }  
});
```

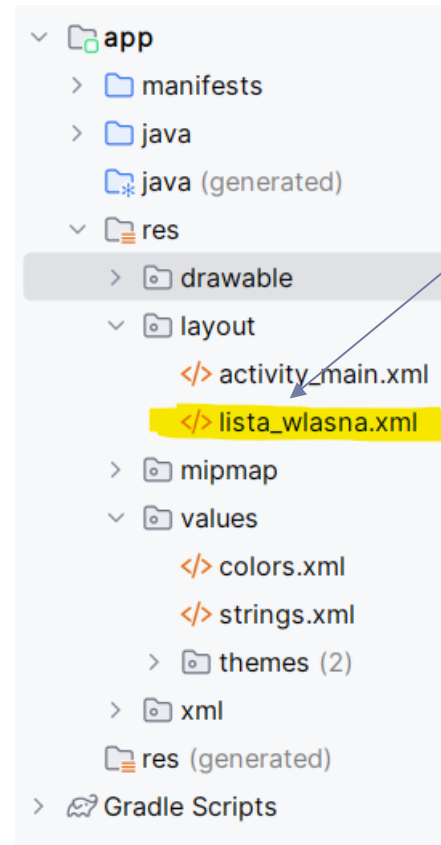
Przypisujemy do listy słuchacz zdarzeń `OnItemClickListener`

W jego metodzie `.onItemClick()` otrzymujemy w parametrze pozycję elementu, na który kliknięto (lub jego id, ale w tym przypadku to jedno i to samo)

## Własny layout listy

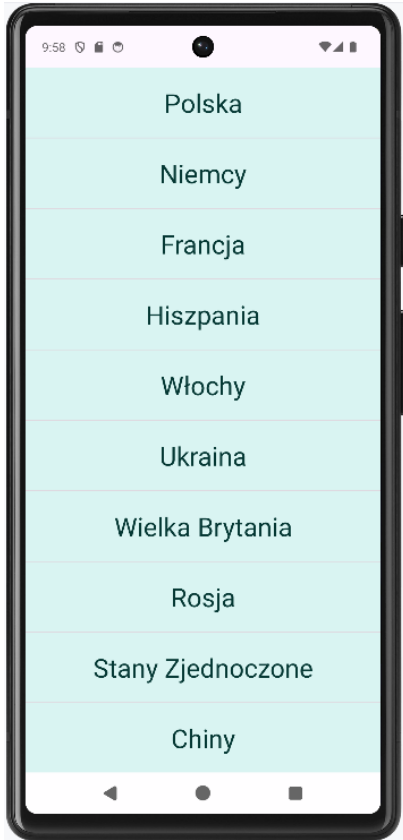
Poprzednim przykładzie korzystaliśmy z standardowego wbudowanego w Androida layoutu listy.

Wygląd elementu listy możemy zdefiniować samodzielnie pamiętać jednak należy, że tworząc prosty layout rezygnujemy z niektórych efektów, na przykład animacji kliknięcia. Stworzenie skomplikowanego layoutu zawierającego np. ikony czy inną grafikę wymaga napisania własnego adaptera.



Pracę rozpoczynamy od stworzenia pliku w folderze res/layout

## Własny layout listy



```
<?xml version="1.0" encoding="utf-8"?>
```

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/element"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="20dp"
    android:textSize="30sp"
    android:textColor="#053834"
    android:background="#D9F4F2"
    android:gravity="center"
/>
```

Plik layoutu zawiera pojedyncze pole tekstowe definiujące wygląd komórki listy.

Szerokość pola powinna być dopasowana do rodzica.  
Wysokość możemy zdefiniować sami.

Wygląd definiujemy podobnie jak dla zwykłego pola tekstowego



Omawiany tu komponent służy do wyświetlania większych zbiorów danych, a umieszczanie takich zbiorów w kodzie języka programowania nie jest dobrą praktyką.

Dane pobierane są zazwyczaj z zasobów aplikacji lub bazy danych.

```
<string-array name="miasta">
  <item>Warszawa</item>
  <item>Berlin</item>
  <item>Paryż</item>
  <item>Madryt</item>
  <item>Rzym</item>
  <item>Londyn</item>
  <item>Moskwa</item>
  <item>Nowy Jork</item>
  <item>Pekin</item>
  <item>Tokio</item>
</string-array>
```

Zamiast tablicy stringów zdefiniowanej w pliku .java możemy taką tablicę umieścić w zasobach aplikacji - w pliku **strings.xml**

## Widok Listy



W kodzie programu tworzymy tablicę i pobieramy do niej dane z zasobów.

```
String[] tablicaMiasta = getResources().getStringArray(R.array.miasta);
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<>(this,  
    android.R.layout.simple_list_item_1, miasta);
```

Lub krócej:

```
ArrayAdapter<String> adapter = new ArrayAdapter<>(this,  
    android.R.layout.simple_list_item_1, getResources().getStringArray(R.array.miasta));
```

