

Ćwiczenia 14 — Android studio – SimpleAdapter, ListView

Na koniec zajęć prześlij pliki źródłowe (.xml, .java)+ obrazek do zasobu w teams.

1. Utwórz projekt o nazwie SimpleAdapter na podstawie Empty Activity, dobierz odpowiednie API.
2. Otworzyć dokumentację:

<https://developer.android.com/reference/android/widget/SimpleAdapter>

<https://developer.android.com/reference/android/widget/ListView#summary>

<https://developer.android.com/reference/android/widget/AdapterView.OnItemClickListener>

<https://developer.android.com/guide/topics/resources/string-resource>

<https://developer.android.com/guide/topics/ui/dialogs>

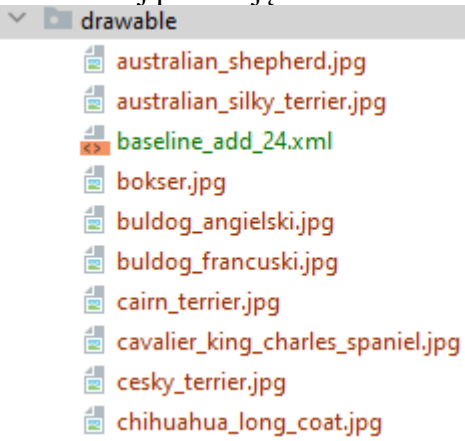
3. Dodaj zależności.
4. Docelowo chcemy uzyskać coś na kształt, wybierz tematykę:



5. Dodaj w activity_main.xml listę

```
<ListView
    android:id="@+id/simpleListView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:divider="#4cAF50"
    android:dividerHeight="3dp"
    android:listSelector="#AC9550"
/>
```

6. Dodaj pliki zdjęć do res ...



7. Nazwy umieść w string-array:

```

1 <resources>
2   <string name="app_name">ListViewDogs</string>
3   <string name="plus">plus</string>
4
5   <string-array name="dogs">
6       <item>australian silky terrier</item>
7       <item>cairn terrier</item>
8       <item>cesky terier</item>
9       <item>chihuahua long coat</item>
10      <item>golden retriever</item>
11      <item>malmut</item>
12      <item>pudel duży</item>
13      <item>cavalier spaniel</item>
14      <item>labrador retriever</item>
15  </string-array>
16
17  <string-array name="dogsDescription">
18      <item>australian silky terrier</item>
19      <item>cairn terrier</item>
20      <item>cesky terier</item>
21      <item>chihuahua long coat</item>
22      <item>golden retriever</item>
23      <item>malmut</item>
24      <item>pudel duży</item>
25      <item>cavalier spaniel</item>
26      <item>labrador retriever</item>
27  </string-array>
28
29 </resources>

```

8. Uzupełnij MainActivity.java:

```

public class MainActivity extends AppCompatActivity {
    2 usages
    private String [] dogs;
    2 usages
    private String [] dogsDesc;
    3 usages
    private Resources resources;
    3 usages
    private ListView listView;
    2 usages
    private FloatingActionButton addItemFAB;
    10 usages
    private HashMap<String, Object> hashMap;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

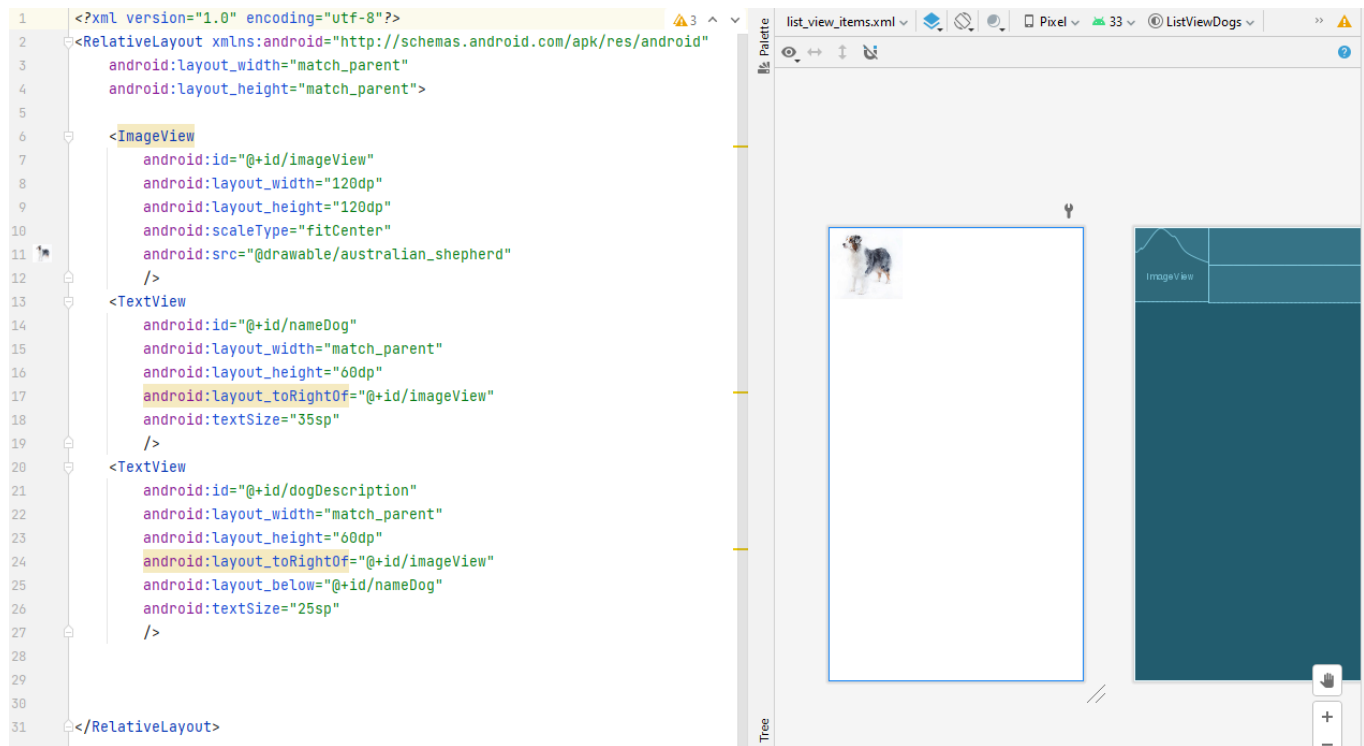
9. Uzupełnij :

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    int [] dogsPictures = {  
        R.drawable.australian_silky_terrier,  
        R.drawable.cairn_terrier,  
        R.drawable.cesky_terrier,  
        R.drawable.chihuahua_long_coat,  
        R.drawable.golden_retriever,  
        R.drawable.malamut,  
        R.drawable.pudel_duzy,  
        R.drawable.cavalier_king_charles_spaniel,  
        R.drawable.labrador_retriever  
    };  
  
    resources = getResources();  
    dogs = resources.getStringArray(R.array.dogs);  
    dogsDesc = resources.getStringArray(R.array.dogsDescription);  
}
```

10. Dodaj elementy do hashMap (metoda put), następnie do listy (metoda add), kod w pętli for możesz zamknąć w metodę, która przyda się przy dodawaniu nowych produktów

```
List<HashMap> dogsList = new ArrayList<>();  
ListView = findViewById(R.id.simpleListView);  
  
for(int i=0; i<dogsPictures.length; i++){  
    HashMap = new HashMap<>();  
    hashMap.put("name", dogs[i]);  
    hashMap.put("image", dogsPictures[i]);  
    hashMap.put("description", dogsDesc[i]);  
    dogsList.add(hashMap);  
}
```

11. Utwórz plik list_view_items.xml:



12. Dodaj do `onCreate()` elementy [from] i [to]

13. Na przykład:

```
String [] from = new String[]{
    "name",
    "image",
    "description"
};
int [] to = new int[]{
    R.id.nameDog,
    R.id.imageView,
    R.id.dogDescription
};
```

14. Stwórz obiekt SimpleAdapter:

```
SimpleAdapter simpleAdapter = new SimpleAdapter(
    getApplicationContext(),
    dogsList,
    R.layout.list_view_items,
    from,
    to
);
```

15. Ustaw adapter zgodnie z dokumentacją:

setAdapter

Added in API level 1

```
public void setAdapter (ListAdapter adapter)
```

Sets the data behind this ListView. The adapter passed to this method may be wrapped by a [WrapperListAdapter](#), depending on the ListView features currently in use. For instance, adding headers and/or footers will cause the adapter to be wrapped.

Parameters**adapter**

ListAdapter: The ListAdapter which is responsible for maintaining the data backing this list and for producing a view to represent an item in that data set.

16. Na przykład:

```
60  
61  
62  
63
```

```
listView.setAdapter(simpleAdapter);
```

17. Wykonaj zadania

a) dodaj obsługę kliknięcia w item listView, wyświetl toast

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
        Toast.makeText(getApplicationContext(), dogsDesc[i], Toast.LENGTH_LONG).show();  
    }  
});
```

b) dodaj obsługę kliknięcia w item ListView, nowa aktywność z opisem pozycji i obrazkiem, zapewnić powrót

7 usages

Zmień deklarację na

`private ArrayList<HashMap<String, Object>> dogsList;`

```

listView.setOnItemClickListener((adapterView, view, i, l) -> {
    Intent intent = new Intent( packageContext: this, PictureActivity.class);
    intent.putExtra( name: "PICTURE_NAME", String.valueOf(dogsList.get(i).get("name")));
    intent.putExtra( name: "PICTURE_DESC", String.valueOf(dogsList.get(i).get("description")));
    intent.putExtra( name: "PICTURE_IMAGE", String.valueOf(dogsList.get(i).get("image")));
    startActivity(intent);
    // Toast.makeText(getApplicationContext(), dogsDesc[i], Toast.LENGTH_LONG).show();
});

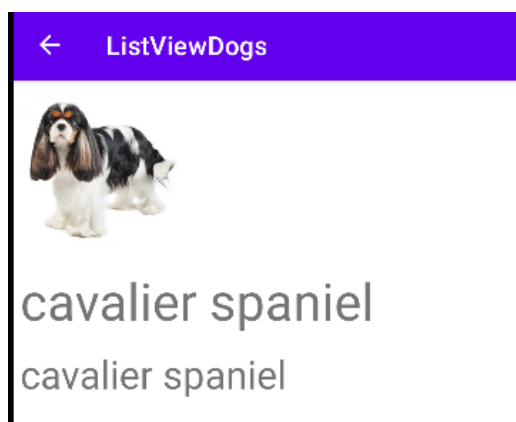
```

3 usages

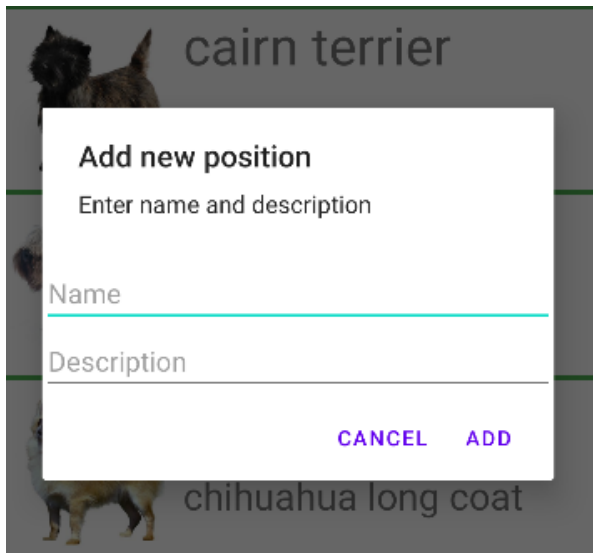
```

9 public class PictureActivity extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_picture);
15
16         ImageView picture = findViewById(R.id.picture);
17         TextView name = findViewById(R.id.name);
18         TextView description = findViewById(R.id.description);
19
20         Bundle bundle = getIntent().getExtras();
21
22         picture.setImageResource(Integer.parseInt(bundle.getString( key: "PICTURE_IMAGE")));
23         name.setText(bundle.getString( key: "PICTURE_NAME"));
24         description.setText(bundle.getString( key: "PICTURE_DESC"));
25     }
26 }

```



c) dodaj opcję dodania nowej pozycji listy poprzez przycisk w kształcie (+)



<https://developer.android.com/guide/topics/ui/dialogs>

```
addItemFAB.setOnClickListener(v->{
    LinearLayout linearLayout = new LinearLayout( context: this);
    linearLayout.setOrientation(LinearLayout.VERTICAL);
    EditText inputName = new EditText( context: this);
    inputName.setHint("Name");
    EditText inputDescription = new EditText( context: this);
    inputDescription.setHint("Description");
    linearLayout.addView(inputName);
    linearLayout.addView(inputDescription);
    AlertDialog dialog = new AlertDialog.Builder( context: this)
        .setTitle("Add new position")
        .setMessage("Enter name and description")
        .setView(linearLayout)
        .setPositiveButton( text: "Add", (dialogInterface, i) ->{
            hashMap = new HashMap<>();
            hashMap.put("name", inputName.getText().toString());
            hashMap.put("description", inputDescription.getText().toString());
            hashMap.put("image", R.drawable.australian_silky_terrier);
            dogsList.add(hashMap);
            listView.setAdapter(simpleAdapter);
        })
        .setNegativeButton( text: "Cancel", listener: null)
        .create();
    dialog.show();
});
```


Kształt dla przycisku:

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <selector xmlns:android="http://schemas.android.com/apk/res/android">
3      <item>
4          <shape android:shape="rectangle">
5              <corners android:radius="100dp">
6              </corners>
7              <solid android:color="#2C7DA5"/>
8              <size android:height="75dp" android:width="75dp" ></size>
9          </shape>
10     </item>
11 </selector>

```

oraz

```

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/addItemFAB"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_margin="16dp"
    android:contentDescription="@string/plus"
    android:src="@drawable/baseline_add_24"
    tools:ignore="ImageContrastCheck" />

```

d) dodaj spinner dla wyboru nazwy obrazka:

```

<Spinner
    android:id="@+id/spinnerImages"
    android:layout_width="match_parent"
    android:tooltipText="Wybierz obrazek"
    android:layout_height="wrap_content"/>

```

I jego obsługa w :

```

addItemFAB.setOnClickListener(v->{
    LinearLayout linearLayout = new LinearLayout( context: this);
    linearLayout.setOrientation(LinearLayout.VERTICAL);
    EditText inputName = new EditText( context: this);

```

```

Spinner imageSpinner = dialogView.findViewById(R.id.spinnerImages);

// Adapter dla spinnera
ArrayAdapter<String> productNameAdapter = new ArrayAdapter<>(
    context: this,
    android.R.layout.simple_spinner_dropdown_item,
    productNames // Używamy nazw produktów, a nie tablicy obrazków!
);

imageSpinner.setAdapter(productNameAdapter);

```

- e) dodaj przycisk usuwający zaznaczony element listy.



Oraz dodaj metodę na kształt:

```

private void removeSelectedItem() {
    if (selectedItem != -1) {
        fruitsList.remove(selectedItem);
        adapter.notifyDataSetChanged();
        selectedItem = -1;
    } else {
        Toast.makeText(context: this, text: "Wybierz element do usunięcia", Toast.LENGTH_SHORT).show();
    }
}

```

- f) zadбай o to, aby nowo dodany element był na liście po obrocie urządzenia o 90 stopni (SharedPreferences)
- <https://developer.android.com/training/data-storage/shared-preferences>
- <https://developer.android.com/reference/android/content/SharedPreferences>
- <https://developer.android.com/develop/ui/views/components/settings/use-saved-values>