

Ćwiczenia 22 — Android studio – Rest API

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

1. Utwórz projekt o nazwie RestApi na podstawie Empty Activity, dobierz odpowiednie API (28 – Android 9).
2. Otwórz dokumentację:

<https://developer.android.com/reference/org/json/JSONArray>

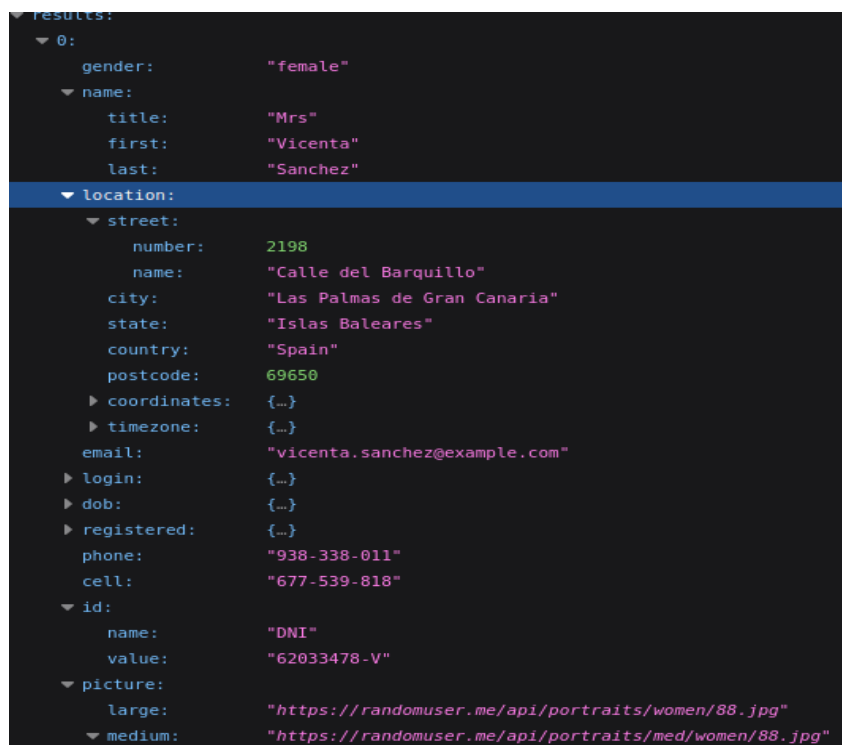
<https://developer.android.com/reference/org/json/JSONObject>

<https://developer.android.com/reference/android/os/AsyncTask>

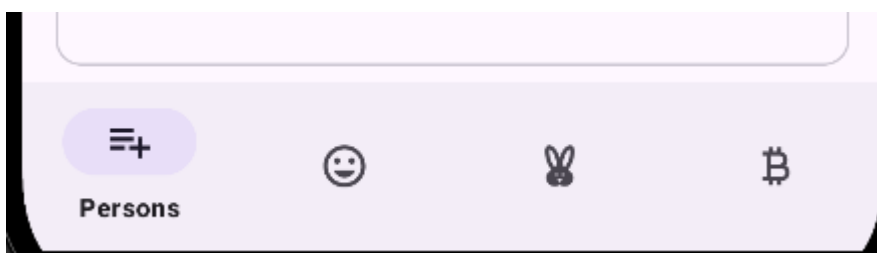
<https://developer.android.com/reference/java/util/concurrent/package-summary>

<https://developer.android.com/reference/java/util/concurrent/Executor>

3. Sprawdź zawartość strony: <https://randomuser.me/api>



4. Stwórz nawigację dolną:



5. Stwórz klasę MyAsyncTask lub klasę implementującą Runnable

```
11 public class MyAsyncTask extends AsyncTask {
12     InputStream inputStream;
13     URL url = null;
14     @Override
15     protected void onPreExecute() {
16         Log.v(AC.TAG, msg: "-----> Start onPreExecute()");
17         super.onPreExecute();
18     }
19
20     @Override
21     protected Object doInBackground(Object[] objects) {
```

6. Zaimplementuj metodę doInBackground()

```
@Override
protected Object doInBackground(Object[] objects) {
    Log.v(AC.TAG, msg: "-----> Start doInBackground()");
    try {
        url = new URL( spec: "https://randomuser.me/api");
    } catch (MalformedURLException e) {
        e.printStackTrace();
    }
    try {
        inputStream = url.openStream();
    } catch (IOException e) {
        e.printStackTrace();
    }
    byte [] buffer = new byte[4096];
    StringBuilder stringBuilder = new StringBuilder("");
    while (true){
        try {
            if (!(inputStream.read(buffer)>0)) break;
        } catch (IOException e) {
            e.printStackTrace();
        }
        stringBuilder.append(new String(buffer));
    }
    publishProgress(stringBuilder.toString());
    return null;
}
```

7. Zaimplementuj pozostałe metody:

```

@Override
protected void onProgressUpdate(Object[] values) {
    Log.v(AC.TAG, msg: "-----> Start onProgressUpdate()");
    Log.v(AC.TAG, values[0].toString());
    super.onProgressUpdate(values);
}

@Override
protected void onPostExecute(Object o) {
    Log.v(AC.TAG, msg: "-----> Start onPostExecute()");
    super.onPostExecute(o);
}

```

8. Wywołaj wątek

```

8 public class MainActivity extends AppCompatActivity {
9     private Button startRandom;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15         startRandom = findViewById(R.id.startApi);
16         startRandom.setOnClickListener(v->{
17             MyAsyncTask myAsyncTask = new MyAsyncTask();
18             myAsyncTask.execute();
19         });
20     }

```

9. Przetestuj aplikację. Oczekiwany efekt:

```

.1: -----> Start onPreExecute()
.1: -----> Start doInBackground()
.1: -----> Start onProgressUpdate()
.1: {"results":[{"gender":"male","name":{"title":"Mr","first":"Biko","last":"Van Kasteren"},"location":{"street":
.1: -----> Start onPostExecute()

```

10. Dodaj:

```

try {
    JSONArray jsonArray = new JSONArray(stringBuilder.toString());
} catch (JSONException e) {
    e.printStackTrace();
}
publishProgress(stringBuilder.toString());
return null;

```

11. Odczytaj składowe

```

JSONObject jsonObject = new JSONObject(stringBuilder.toString());
JSONArray jsonArray = jsonObject.getJSONArray( name: "results");
JSONObject person = jsonArray.getJSONObject( index: 0);
JSONObject name = person.getJSONObject("name");
String strname = name.getString( name: "title")
    +". " +name.getString( name: "first")
    +". " +name.getString( name: "last");
String email = person.getString( name: "email");
String gender = person.getString( name: "gender");

publishProgress( ...values: strname,0);
publishProgress( ...values: email,1);
publishProgress( ...values: gender,2);
}catch (JSONException e) {

```

12. Uzupełnij metodę onProgressUpdate

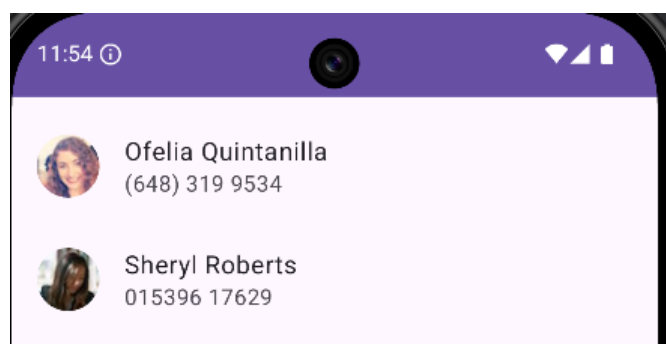
```

@Override
protected void onProgressUpdate(Object[] values) {
    Log.v(AC.TAG, msg: "-----> Start onProgr
    Log.v(AC.TAG,values[0].toString());
    Context context;

    Log.v(AC.TAG, msg: "values[1]="+values[1]);
    switch (Integer.parseInt( s: values[1]+"")){
        case 0:
            name.setText("Name: " + values[0].toString());
            break;
        case 1:
            email.setText("Email: " + values[0].toString());
            break;
        case 2:
            gender.setText("Gender: " + values[0].toString());
            break;
    }
}

```

13. Przetestuj aplikację:



14. Krok 3: załaduj zdjęcie: <https://developer.android.com/topic/performance/graphics/load-bitmap>
<https://square.github.io/picasso/>

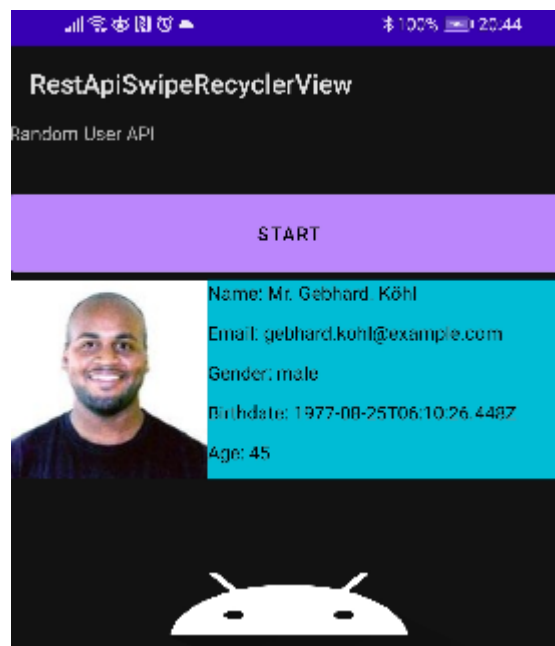
GRADLE

```
implementation 'com.squareup.picasso:picasso:(insert latest version)'
```

15. Załaduj zdjęcie:

```
case 2:
    gender.setText("Gender: " + values[0].toString());
    break;
case 3:
    Picasso.get().load(values[0].toString()).into(imagePerson);
    break;
```

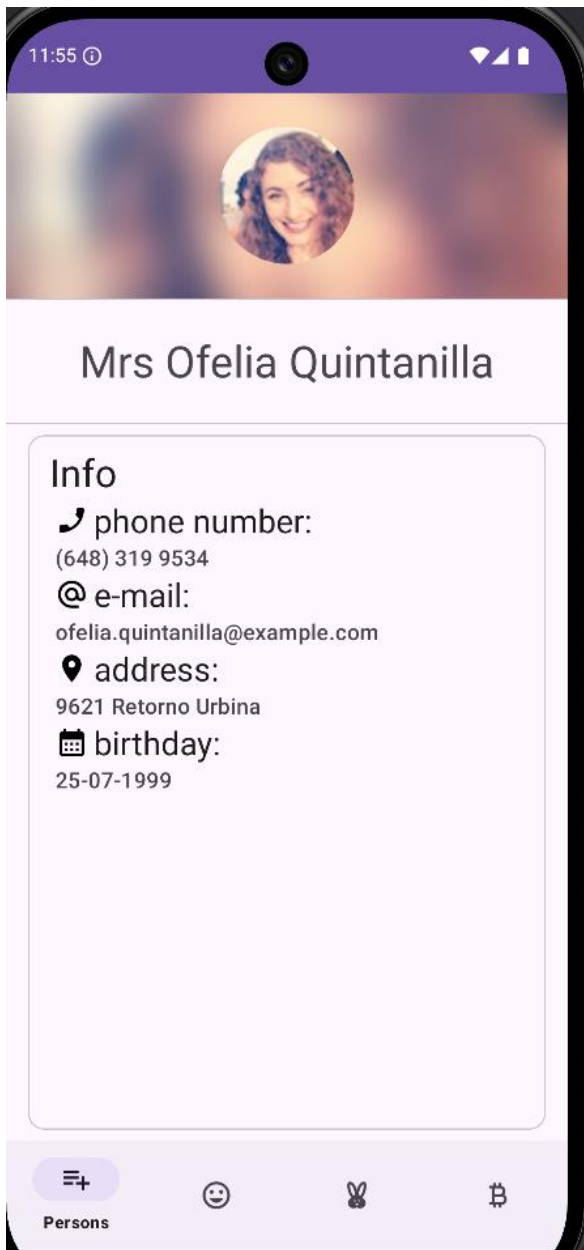
16. Dodaj datę urodzenia oraz wiek tak, aby otrzymać:



17. Samodzielnie dodaj pola: telefon, państwo, miasto i ulicę.

18. Umieść dane osób na komponencie ListView lub RecyclerView, kliknięcie pozycji listy przenosi do nowej aktywności na której wyświetlisz zdjęcie i dane osoby

19. Po kliknięciu w pozycję listy otwiera się nowy widok:



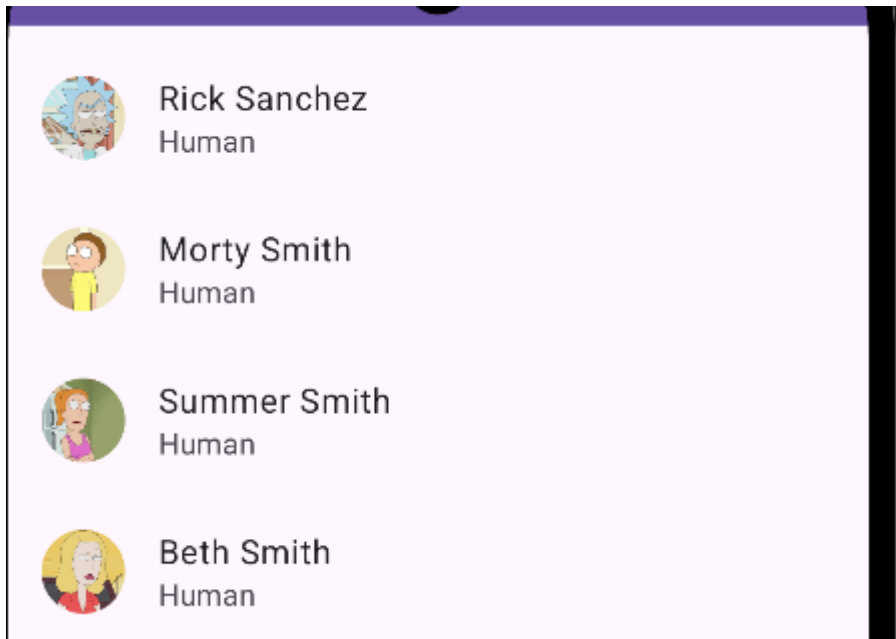
20. Wykonaj:

a) dodaj zmianę wartości, zdjęcia poprzez odświeżenie `SwipeRefreshLayout` **swipeRefreshLayout**;

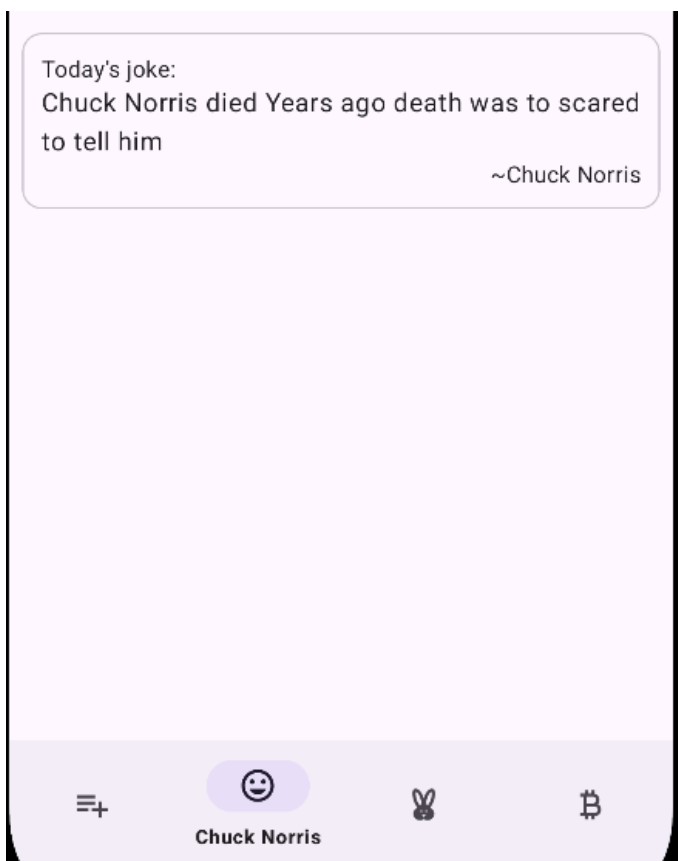
zastąp zdarzenie kliknięcia w przycisk na przeciągnięcie palcem

swipeRefreshLayout.setOnRefreshListener() -> {

b) utwórz aktywność dla innego API np.:



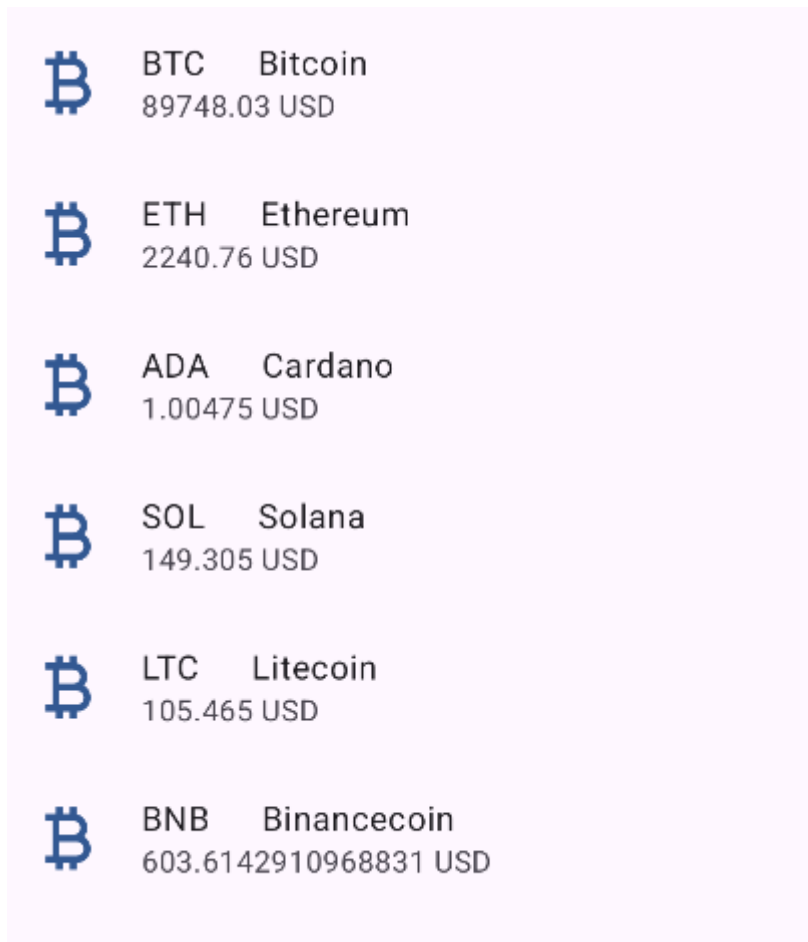
<https://rickandmortyapi.com/api/character/> lub



<https://api.chucknorris.io/jokes/random> lub

<https://api.fbi.gov/wanted/v1/list> lub

<https://api.coinbase.com/v2/currencies/crypto>



c) zbuduj własne API: <https://www.postman.com/>

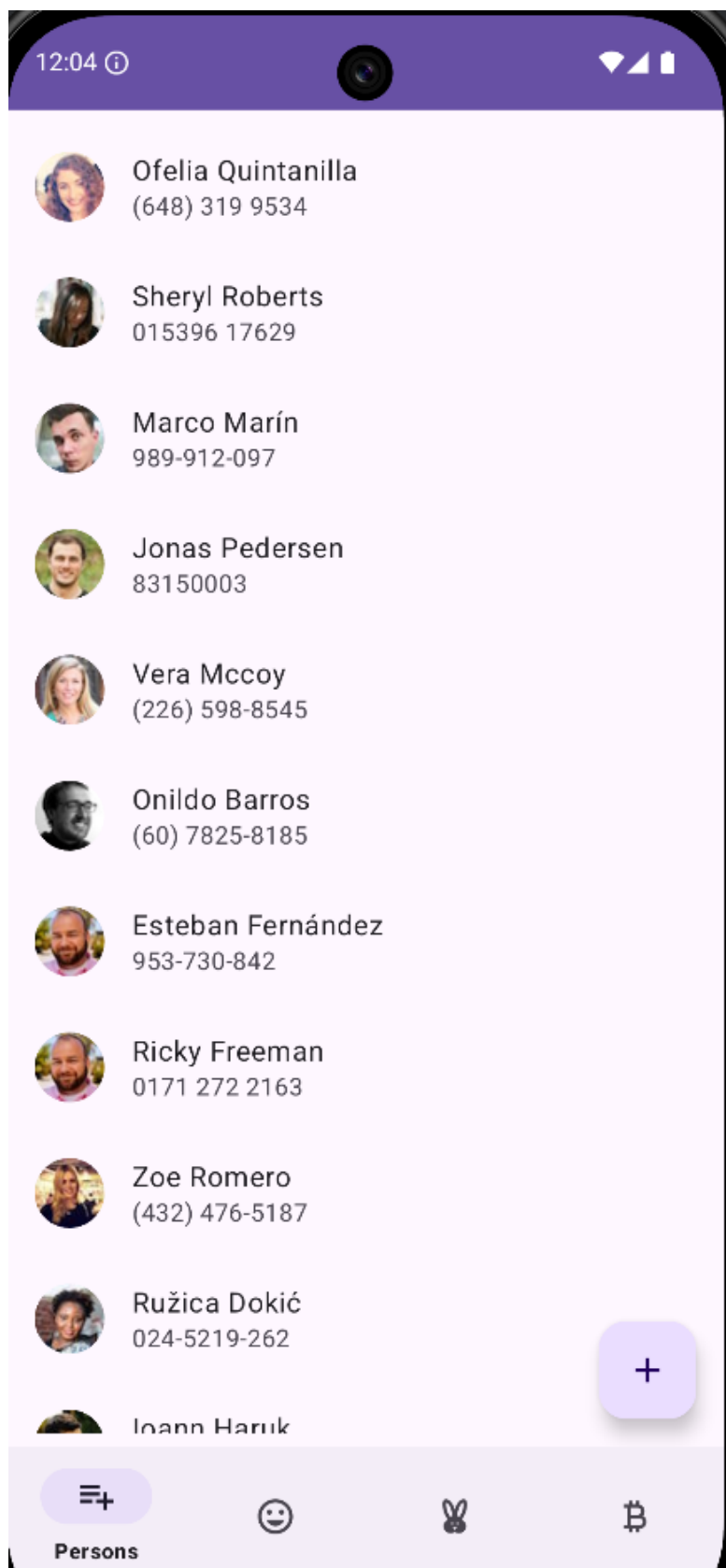
d) skorzystaj z innej biblioteki niż Picasso

e) zrealizuj ćwiczenie w oparciu o:

<https://developer.android.com/reference/java/util/concurrent/package-summary>

<https://developer.android.com/reference/java/util/concurrent/Executor>

21. Widok całej aplikacji:



22. KONIEC.