



# Komunikacja HTTP (I)

Marek Wojtuszkiewicz

# API

Interfejs programistyczny aplikacji (ang. **Application Programming Interface**, API) – sposób, rozumiany jako ściśle określony zestaw reguł i ich opisów, w jaki programy komputerowe komunikują się między sobą. API definiuje się na poziomie kodu źródłowego dla takich składników oprogramowania jak np. aplikacje, biblioteki czy system operacyjny. Zadaniem API jest dostarczenie odpowiednich specyfikacji podprogramów, struktur danych, klas obiektów i wymaganych protokołów komunikacyjnych.

źródło: [https://pl.wikipedia.org/wiki/Application\\_Programming\\_Interface](https://pl.wikipedia.org/wiki/Application_Programming_Interface)



## API – przykładowe źródła danych

<https://www.publicapis.com>

<https://github.com/toddmotto/public-apis>

<https://www.data.gov>

<https://www.programmableweb.com/apis/directory>

<https://developers.google.com/apis-explorer>



# JSON

JavaScript Object Notation\*

Popularny format wymiany danych

\* nie należy sugerować się tą nazwą





## Struktura JSON

- JSON Object
- JSON Array
- JSON Primitive

```
{"nazwa_kursu":"Java -  
Android","edycja":2,"tematy":["Wprowadzenie","Material  
Design","Komunikacja HTTP"], "sub_object": {"property":"value",  
"another_sub_object":{"array":["value_1","value_2"],"number":1,"bo  
olean":true}}, "uczestnicy":[{"id":1,"imie":"Jan","nazwisko":"Kowalski"  
}, {"id":2,"imie":"Johnny","nazwisko":"English"}]}
```



# Struktura JSON

- JSON Object
- JSON Array
- JSON Primitive

```
{
  "nazwa_kursu": "Java - Android",
  "edycja": 2,
  "tematy": [
    "Wprowadzenie",
    "Material Design",
    "Komunikacja HTTP"
  ],
  "sub_object": {
    "property": "value",
    "another_sub_object": {
      "array": [
        "value_1",
        "value_2"
      ],
      "number": 1,
      "boolean": true
    }
  },
  "uczestnicy": [
    {
      "id": 1,
      "imie": "Jan",
      "nazwisko": "Kowalski"
    },
    {
      "id": 2,
      "imie": "Johnny",
      "nazwisko": "English"
    }
  ]
}
```



# Obsługa formatu JSON w systemie Android

JSONObject

JSONArray

JSONException

```
try {  
    JSONObject jsonObject = new JSONObject(data);  
  
    JSONObject subObject = jsonObject.getJSONObject("sub_object");  
    String property = subObject.getString("value");  
  
    JSONArray uczestnicy = jsonObject.getJSONArray("uczestnicy");  
  
    int ilosc_uczestnikow = uczestnicy.length();  
  
    String imie_pierwszego_uczestnika =  
        uczestnicy.getJSONObject(0).getString("imie");  
  
    String imie_pierwszego_uczestnika_2 =  
        uczestnicy.getJSONObject(0).optString("imie");  
} catch (JSONException e) {  
    Log.d(TAG, "Wystąpił błąd przy odczycie danych: " + e.getMessage());  
}
```

## JSON - linki

<http://www.json.org>

<http://jsonprettyprint.com>

<https://jsonformatter.curiousconcept.com>

<http://jsonmate.com>





## Warunki dla nawiązania komunikacji HTTP w systemie Android

- wymaga przyznania uprawnienia `android.permission.INTERNET`
- nie można nawiązywać komunikacji w wątku głównym aplikacji (wątku UI) - `NetworkOnMainThreadException`





# URLConnection

1. Otwiera/zamyka połączenie HTTP
2. Formuje zapytanie do serwera
3. Przesyła zapytanie do serwera
4. Odbiera odpowiedź z serwera

```
try {
    URL url = new URL("http://sdacademy.pl");
    urlConnection = (URLConnection) url.openConnection();

    InputStream inputStream = new BufferedInputStream(urlConnection.getInputStream());

    StringBuilder textBuilder = new StringBuilder();
    try (Reader reader = new BufferedReader(new InputStreamReader(inputStream,
        Charset.forName(StandardCharsets.UTF_8.name())))) {
        int c;
        while ((c = reader.read()) != -1) {
            textBuilder.append((char) c);
        }
    }

    String response = textBuilder.toString();

} catch (IOException e) {
    e.printStackTrace();
} finally {
    urlConnection.disconnect();
}
```



# AsyncTask

Abstrakcyjna klasa pozwalająca na wykonanie czasochłonnych zadań poza wątkiem UI aplikacji (doInBackground) lub w wątku UI aplikacji (onPreExecute, onProgressUpdate, onPostExecute)

```
private class GetDataTask extends AsyncTask<String, Void, String> {  
  
    @Override  
    protected String doInBackground(String... args) {  
        String url = args[0];  
        String data = getData(url);  
        return data;  
    }  
  
    @Override  
    protected void onPostExecute(String data) {  
        textView.setText(data);  
    }  
}
```

```
new GetDataTask().execute("http://sdacademy.pl");
```



# Komunikacja HTTP (I) - zadanie

Utwórz aplikację służącą wyświetlaniu informacji o aktualnej pogodzie w wybranym przez Ciebie miejscu.

1. użyj OpenWeatherMap API (dokumentacja znajduje się pod adresem <https://openweathermap.org/api>)
2. W UI aplikacji wyświetl wszystkie dostępne dane (wskazówka: projektując interface użytkownika możesz posłużyć się zasadami Material Design)

Opcjonalnie:

1. Dodaj informację o długości dnia
2. Zmodyfikuj aplikację w taki sposób, aby wskazywała aktualną pogodę dla dowolnego miejsca na świecie



## Bonus: biblioteka joda-time

Biblioteka wspomagająca  
obsługę daty czasu

Funkcjonalność biblioteki  
została w dużej mierze  
przeniesiona do Java 8 Date  
and Time API (JSR-310)

<http://www.joda.org/joda-time>

```
// data i czas
long dateAsMilliseconds = 1483225200000L;
DateTime someDateAndTime = new DateTime(dateAsMilliseconds);
someDateAndTime.toString("HH:mm");

//czas trwania
Period period = new Period(startTime, endTime);
PeriodFormatter formatter = new PeriodFormatterBuilder()
    .appendHours().appendSeparator(":").appendMinutes().toFormatter();
period.toString(formatter);
```



## Bonus: biblioteka Apache Commons IO

Biblioteka zawiera zestaw narzędzi ułatwiających pracę z funkcjonalnością IO, m.in. strumieniami danych.

Przed:

```
StringBuilder textBuilder = new StringBuilder();
try (Reader reader = new BufferedReader(
    new InputStreamReader(stream, Charset.forName(StandardCharsets.UTF_8.name())))) {
    int c;
    while ((c = reader.read()) != -1) {
        textBuilder.append((char) c);
    }
}

String output = textBuilder.toString();
```

Po:

```
String output = IOUtils.toString(stream, StandardCharsets.UTF_8.name());
```