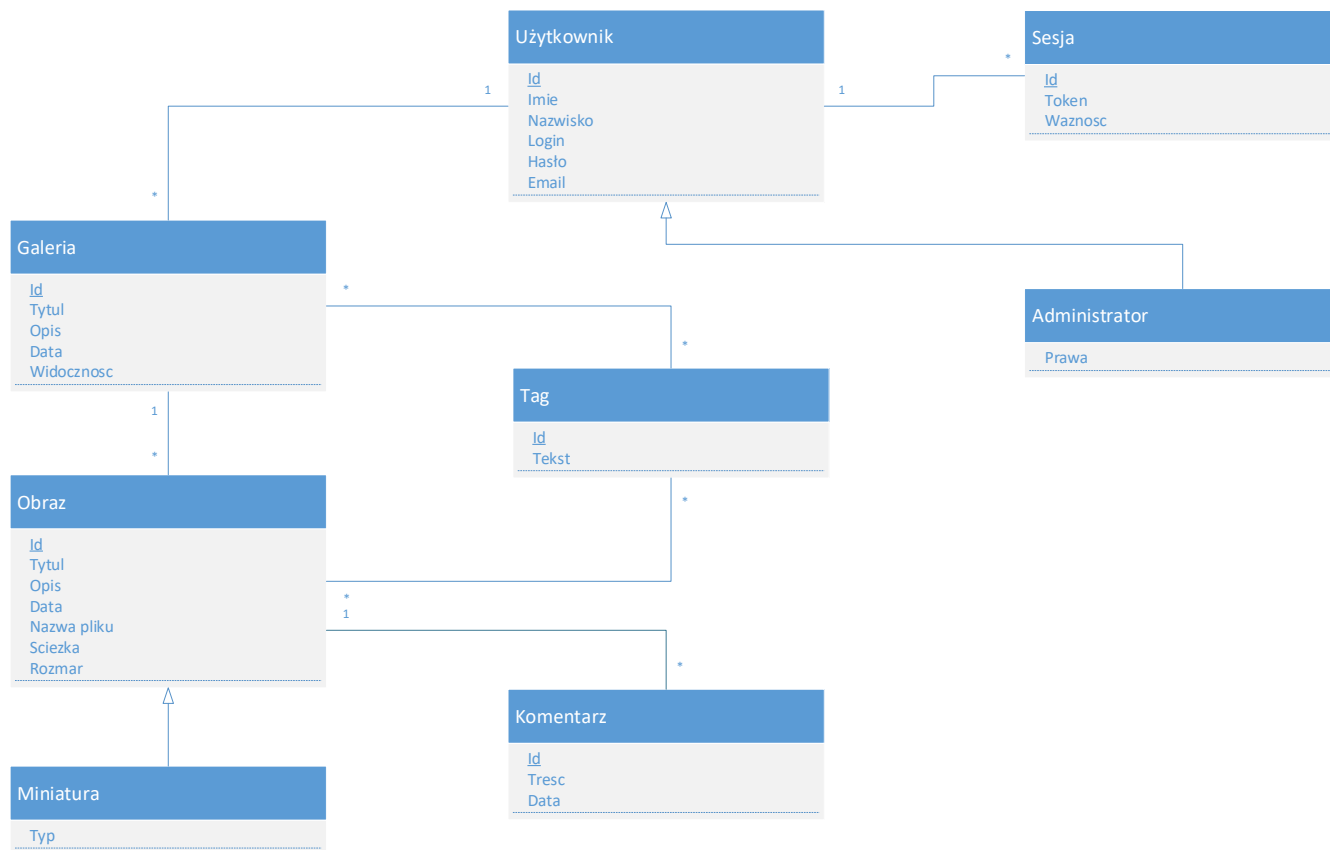


DEFINIOWANIE SCHEMATÓW DANYCH W SWAGGER



TYPY DANYCH W DEFINICJACH SWAGGERA

OpenAPI, na którym opiera się Swagger definiuje następujące podstawowe typy danych:

- string
- number
- integer
- boolean
- array
- object

W przypadku konieczności skorzystania z odpowiedzi typu *mixed type* stosuje się słowo kluczowe *oneOf* lub *anyOf* zamiast *type*.

Dla typu liczbowego *number* zdefiniowano następujące dopuszczalne formaty:

- bez formatu – dowolna liczba
- float
- double

Dla typu liczbowego *integer* zdefiniowano następujące dopuszczalne formaty:

- bez formatu – dowolna liczba całkowita
- int32
- int64

Dla liczb możemy też zdefiniować wartości *min* i *max*, oraz mnożnik *multipleOf* (na przykład dla 10 będą to wartości takie jak: -20, -10, 0, 10 itede).

W przypadku łańcuchów znaków możliwe jest określenie minimalnej i maksymalnej długości (*minLength*, *maxLength*) i wzorca w postaci wyrażenia regularnego (*pattern*).

Formaty dostępne dla typu łańcuchowego obejmują:

- date – data zdefiniowana jak w RFC 3339
- date-time data i czas zdefiniowana jak w RFC 3339 na przykład: 2017-07-21T17:32:28Z
- password – hasło
- byte – znaki kodowane base64-encoded
- binary – dane binarne (pliki)
- email
- uuid
- uri
- hostname
- ipv4
- ipv6

Tablice definiuje się jak następuje i mogą być zagnieżdżane oraz zawierać obiekty:

```
type: array
```

```
items:
  type: string
  type: array
  items:
    type: object
    properties:
      id:
        type: integer
```

Możemy też zdefiniować minimalną (*minItems*) i maksymalną (*maxItems*) liczbę elementów jak i ich unikalność (*uniqueItems: true*).

W przypadku definiowania obiektów musimy opisać wszystkie jego składowe wraz ze zdefiniowanym typem oraz wskazania pól obowiązkowych.

```
type: object
properties:
  id:
    type: integer
    readOnly: true
  username:
    type: string
  name:
    type: string
required:
  - id
  - username
```

ZADANIE

W oparciu o diagram klas korzystając ze Swaggera zaprojektuj poszczególne schematy danych. Przykładowo dla zwrócenia wyniku pobrania aktywnej sesji może to wyglądać tak:

```
48 ▾ definitions:
49 ▾   SesjaResponse:
50 ▾     required:
51 ▾       - Id
52 ▾       - Token
53 ▾       - Waznosc
54 ▾     properties:
55 ▾       Id:
56 ▾         type: string
57 ▾       Token:
58 ▾         type: string
59 ▾       Waznosc:
60 ▾         type: string
61 ▾         format: date-time
```

Responses

Code	Description	Schema
200	Success	<pre>↔ ▼ SesjaResponse { Id: string * Token: string * Waznosc: ▼ string * (date-time) }</pre>